



TUGAS AKHIR - TM 141585

RANCANG BANGUN *SELF-BALANCING* PADA *INVERTED PENDULUM* MENGGUNAKAN *CONTROL MOMENT GYROSCOPE* DENGAN *DOUBLE GYROSCOPE* DAN SISTEM KENDALI PID

MOCH YUSRON NAFI'
NRP 2113100028

Dosen Pembimbing
Unggul Wasiwitono, ST., M.Eng.Sc., Dr.Eng.

DEPARTEMEN TEKNIK MESIN
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - TM141585

RANCANG BANGUN *SELF-BALANCING* PADA *INVERTED PENDULUM* MENGGUNAKAN *CONTROL MOMENT GYROSCOPE* DENGAN *DOUBLE GYROSCOPE* DAN SISTEM KENDALI PID

Moch Yusron Nafi'
NRP 2113100028

Dosen Pembimbing
Unggul Wasiwitono, ST., M.Eng.Sc., Dr.Eng.

DEPARTEMEN TEKNIK MESIN
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017

Halaman ini sengaja dikosongkan



FINAL PROJECT - TM141585

**DESIGN AND IMPLEMENTATION OF SELF-BALANCING
ON INVERTED PENDULUM USING CONTROL MOMENT
GYROSCOPE WITH DOUBLE GYROSCOPE AND PID
CONTROLLER**

Moch Yusron Nafi'
NRP 2113100028

Advisor
Unggul Wasiwitono, ST., M.Eng.Sc., Dr.Eng.

DEPARTMENT OF MECHANICAL ENGINEERING
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2017

Halaman ini sengaja dikosongkan

**RANCANG BANGUN SELF-BALANCING PADA INVERTED
PENDULUM MENGGUNAKAN CONTROL MOMENT
GYROSCOPE DENGAN DOUBLE GYROSCOPE DAN SISTEM
KENDALI PID**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Program Studi S-1 Departemen Teknik Mesin
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh :

MOCH YUSRON NAFI

NRP. 2113 100 028

Disetujui oleh Tim Penguji Tugas Akhir :

- Unggul Wasiwitono, ST., M.Eng.Sc., Dr.Eng. (Pembimbing)
NIP. 197805102001121001
- Prof. Ir. I Nyoman Sutantra, MSc.PhD (Penguji I)
NIP. 195106051978031002
- Ari Kurniawan Saputra, ST., MT. (Penguji II)
NIP. 198604012015041001



**SURABAYA
JULI, 2017**

a

Halaman ini sengaja dikosongkan

ABSTRAK

Nama : Moch Yusron Nafi'
Judul : Rancang Bangun *Self-balancing* Pada *Inverted Pendulum* Menggunakan *Control Moment Gyroscope* Dengan *Double Gyroscope* Dan Sistem Kendali PID.
Pembimbing : Unggul Wasiwitono, ST., M.Eng.Sc., Dr.Eng.

Perkembangan teknologi saat ini berjalan sangat pesat, salah satunya yaitu perkembangan teknologi giroskop. Giroskop merupakan teknologi yang lazim digunakan untuk kendali pesawat ruang angkasa dan satelit. Selain itu giroskop juga dikembangkan dan diterapkan pada kendaraan untuk meningkatkan kestabilan. Penelitian ini menggunakan prinsip *control moment gyroscope* untuk menstabilkan sebuah *inverted pendulum*. Oleh karena itu, diperlukan alat peraga untuk melakukan penelitian tersebut. Penelitian ini merancang dan membuat alat peraga berupa sistem *inverted pendulum double gyroscope* menggunakan prinsip *control moment gyroscope* untuk menyeimbangkan diri secara otomatis.

Penelitian ini menggunakan analisa teoritis dengan membuat persamaan matematis dari alat peraga yang akan dikendalikan. Sedangkan sistem kendali yang digunakan adalah PID. Sistem kendali PID digunakan untuk memperbaiki respon sistem dan menghilangkan eror.

Hasil yang didapatkan dari penelitian ini adalah harga parameter $P=7,33$, $I=5,33$, dan $D=1,2$ yang merupakan harga paling optimum untuk sistem *inverted pendulum* berdasarkan eksperimen. Alat penelitian hanya dapat mempertahankan keseimbangan selama delapan hingga sepuluh detik diakibatkan oleh pemodelan yang diterapkan linearisasi dan mekanik alat yang kurang presisi.

Kata kunci: giroskop, *inverted pendulum*, *control moment gyroscope*, PID.

Halaman ini sengaja dikosongkan

ABSTRACT

Name : Moch Yusron Nafi'
Title` : *Design And Implementation Of Self-balancing on Inverted Pendulum Using Control Moment Gyroscope With Double Gyroscope And PID Controller.*
Advisor : Unggul Wasiwitono, ST., M.Eng.Sc., Dr.Eng.

The development of technology is currently running rapidly, including the development of gyroscope technology. Gyroscope is a technology commonly used for the control of spaceships and satellites. In addition, gyroscopes are also developed and applied to vehicles to improve stability. This study uses the principle of control moment gyroscope to stabilize an inverted pendulum. Therefore, the prototype needed to do this research. This research designs and makes prototype in the form of double gyroscope inverted pendulum using control moment gyroscope principle to automatic self-balancing the system.

This study used theoretical analysis by modeling equation of motion from the prototype. The controller used by this project is PID controller. PID controller is used to improve transient response and steady state error.

The result of this study gives us the control value of $P=7.33$, $I=5.33$, and $D=1.2$. This control parameter is pretty effective to balance the pendulum in this experiment. This controller could balance the system for eight to ten second. This limited time caused by linearization that making this controller has small workspace in small angle. The prototype also manufactured by manual machine which is not too good for precision result.

Keyword: *gyroscope, control moment gyroscope, PID.*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT atas segala limpahan berkah, rahmat, serta hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul Rancang Bangun *Self-balancing* Pada *Inverted Pendulum* Menggunakan *Control Moment Gyroscope* Dengan *Double Gyroscope* Dan Sistem Kendali PID. Penelitian ini disusun dalam rangka pemenuhan riset di Jurusan Teknik Mesin ITS, bidang studi desain, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak dan Saudara tercinta yang telah memberikan dorongan spiritual dan material dalam penyelesaian penelitian ini.
2. Bapak Unggul Wasiwitono, ST., M.Eng.Sc., Dr.Eng., selaku dosen pembimbing yang senantiasa mengarahkan serta memberi koreksi pada setiap kesalahan pengerjaan.
3. Bapak/Ibu dosen, karyawan, serta seluruh keluarga besar Teknik Mesin ITS.
4. Seluruh teman-teman laboratorium otomotif Teknik Mesin ITS.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Juli 2017

Halaman ini sengaja dikosongkan

DAFTAR ISI

| | |
|--|------|
| LEMBAR PENGESAHAN | i |
| ABSTRAK..... | iii |
| ABSTRACT..... | v |
| KATA PENGANTAR | vii |
| DAFTAR ISI..... | ix |
| DAFTAR GAMBAR | xi |
| DAFTAR TABEL..... | xiii |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 1 |
| 1.3 Batasan Masalah | 2 |
| 1.4 Tujuan Penelitian | 2 |
| 1.5 Manfaat Penelitian | 2 |
| BAB II TINJAUAN PUSTAKA..... | 3 |
| 2.1 Penelitian Terdahulu | 3 |
| 2.2 Girooskop | 3 |
| 2.3 <i>Control Moment Gyroscope</i> | 4 |
| 2.4 Sistem kendali PID | 6 |
| 2.4.1 Kontrol Proporsional | 6 |
| 2.4.2 Kontrol Integratif..... | 6 |
| 2.4.3 Kontrol Derivatif | 6 |
| BAB III METODOLOGI PENELITIAN..... | 9 |
| 3.1 Diagram Alir Penelitian Penelitian | 9 |
| 3.2 Langkah–langkah Penelitian..... | 10 |
| BAB IV ANALISA DAN PEMBAHASAN..... | 17 |
| 4.1 Pemodelan dan Penentuan Spesifikasi Alat | 17 |
| 4.1.1 Konfigurasi penyeimbang girooskop..... | 17 |
| 4.1.2 <i>Free body diagram inverted pendulum</i> | 19 |
| 4.1.3 Penentuan Spesifikasi Elektronik | 20 |
| 4.2. Blok Diagram Pengendalian | 22 |
| 4.3 <i>Complementary Filter</i> | 23 |
| 4.3.1 Masalah pada <i>accelerometer</i> | 23 |
| 4.3.2 Masalah pada <i>gyroscope</i> | 24 |
| 4.3.3 Filter | 24 |
| 4.4 Sistem Identifikasi Servo | 25 |
| 4.5 Analisa Stabilitas | 29 |

| | |
|-------------------------------------|----|
| 4.6 Desain Sistem Kendali PID | 30 |
| 4.7 Eksperimen | 33 |
| BAB V | 37 |
| 5.1 Kesimpulan..... | 37 |
| 5.2 Saran..... | 37 |
| DAFTAR PUSTAKA | 38 |
| LAMPIRAN | 39 |
| BIOGRAFI PENULIS | 49 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 Arah momentum sudut ($I\omega$), kecepatan putaran gimbal $\dot{\alpha}$, dan torsi giroskopik τ_g | 5 |
| Gambar 3. 1 Diagram alir penelitian | 9 |
| Gambar 3. 2 Diagram desain PID <i>controller</i> | 11 |
| Gambar 3. 3 Diagram alir perancangan dan pembuatan <i>prototype</i> | 12 |
| Gambar 3. 4 Diagram alir pembuatan <i>controller</i> | 13 |
| Gambar 3. 5 Skema elektronik pengendalian motor servo | 15 |
| Gambar 3. 6 Skema elektronik motor brushless DC | 16 |
| Gambar 4. 1 Konfigurasi <i>flywheel</i> pada <i>inverted pendulum</i> dengan <i>double gyroscope</i> | 17 |
| Gambar 4. 2 Gaya-gaya yang bekerja pada <i>inverted pendulum</i> | 19 |
| Gambar 4. 3 Blok diagram pengendalian keseimbangan <i>inverted pendulum</i> | 22 |
| Gambar 4. 4 Data sensor IMU tanpa filter | 24 |
| Gambar 4. 5 Data sensor IMU menggunakan <i>Complementary filter</i> .. | 25 |
| Gambar 4. 6 <i>Step response</i> motor servo pada setpoint 10° | 26 |
| Gambar 4. 7 <i>Step response</i> motor servo pada setpoint 20° | 26 |
| Gambar 4. 8 <i>Step response</i> motor servo pada setpoint 30° | 27 |
| Gambar 4. 9 <i>Root locus</i> dari sistem <i>inverted pendulum</i> | 29 |
| Gambar 4. 10 <i>Root locus</i> dari sistem <i>inverted pendulum</i> dengan kompensasi zero pada $s = -7,0036$ | 30 |
| Gambar 4. 11 <i>Step response inverted pendulum</i> | 31 |
| Gambar 4. 12 <i>Step response</i> dari dua nilai kendali PD | 32 |
| Gambar 4. 13 <i>Step response</i> sistem <i>inverted pendulum</i> dengan kendali PD dan PID | 33 |
| Gambar 4. 14 Grafik sudut <i>roll</i> θ terhadap waktu | 34 |
| Gambar 4. 15 Grafik sudut α terhadap waktu | 35 |

Halaman ini sengaja dikosongkan

DAFTAR TABEL

| | |
|---|----|
| Tabel 3. 1 Komponen mekanik dan elektronik penelitian..... | 15 |
| Tabel 4. 1 Torsi giroskopik τ_g terhadap variasi kecepatan <i>flywheel</i> ω | 18 |
| Tabel 4. 2 Torsi <i>inverted pendulum</i> τ_p terhadap variasi sudut kemiringan θ | 21 |
| Tabel 4. 3 Nilai parameter percobaan | 23 |
| Tabel 4. 4 Nilai Parameter <i>Transient Response</i> | 28 |
| Tabel 4. 5 Nilai ω_n dan ζ untuk tiap-tiap setpoint | 28 |

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kendaraan roda dua atau sepeda motor merupakan kendaraan yang paling banyak dimiliki masyarakat Indonesia. Sepeda motor dipilih masyarakat karena memiliki harga yang relatif lebih murah dibanding mobil serta memiliki mobilitas yang tinggi terutama terutama di perkotaan yang sering mengalami kemacetan. Jumlah sepeda motor per maret 2017 di Indonesia adalah sebanyak 83.088.693 unit [6]. Sedangkan data penjualan sepeda motor untuk tahun 2016 adalah sebesar 5.931.285 unit, artinya terdapat penambahan jumlah sepeda motor per tahun hampir sebesar 6 juta unit [5].

Disamping menjadi alat transportasi yang paling digemari, sepeda motor ternyata juga memiliki tingkat resiko kecelakaan yang tinggi. Jumlah sepeda motor yang mengalami kecelakaan lalu lintas pada semester dua tahun 2016 adalah sebesar 70.785 unit [5]. Sepeda motor mudah mengalami kecelakaan jika dibandingkan dengan kendaraan beroda empat atau lebih. Hal ini disebabkan karena pada dasarnya kendaraan roda dua itu tidak stabil. Solusi dari permasalahan ini adalah dengan menerapkan efek giroskop untuk dapat menunjang kestabilan dari kendaraan tersebut. Penelitian kali ini dilakukan untuk merancang penyeimbang berbasis giroskop. Model prototype yang digunakan adalah berupa *inverted pendulum* sebagai model dari sebuah kendaraan.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa permasalahan, antara lain:

1. Bagaimana cara merancang *self-balancing inverted pendulum* dengan menggunakan penyeimbang giroskop?
2. Bagaimana cara menyeimbangkan *inverted pendulum* dengan menggunakan penyeimbang giroskop?
3. Bagaimana cara membuat sistem kendali penyeimbang *inverted pendulum*?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Sistem penyeimbang yang digunakan menggunakan prinsip *control moment gyroscope*.
2. Sistem kendali yang digunakan adalah PID.
3. Penelitian kestabilan dilakukan hanya pada sudut *rolling*.
4. Pengujian dilakukan pada keadaan permukaan yang datar

1.4 Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah:

1. Merancang *self-balancing inverted pendulum* dengan menggunakan penyeimbang giroskop?
2. Menyeimbangkan *inverted pendulum* dengan menggunakan penyeimbang giroskop.
3. Membuat sistem kendali untuk menyeimbangkan *inverted pendulum*.

1.5 Manfaat Penelitian

Adapun manfaat yang didapat dari penelitian ini adalah:

1. Mengembangkan teknologi penyeimbangan yang dapat digunakan untuk meningkatkan stabilitas kendaraan.
2. Mengurangi tingkat resiko kecelakaan pada kendaraan yang diaplikasikan penyeimbang giroskop.
3. Menciptakan *prototype* yang dapat digunakan sebagai bahan ajar di kelas atau laboratorium.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Giroskop mulai dikembangkan pada dunia industri sejak awal abad ke-20 masehi untuk menunjang kestabilan monorail. Setelah itu, giroskop dikembangkan lagi sebagai kendali satelit dan pesawat luar angkasa. Pada awal tahun 2000-an, giroskop mulai dikembangkan lagi untuk menunjang kestabilan kendaraan, terutama kendaraan roda. Hal ini disebabkan karena pada dasarnya kendaraan roda tidak stabil.

Penelitian terdahulu yang menjadi rujukan penulis yaitu sebuah penelitian *thesis* milik *walck* (2013) [8]. Dalam penelitiannya *walck* memodelkan *inverted pendulum* kedalam model matematis, lalu melakukan analisa kestabilan terhadap sistem. Hasil yang didapatkan yakni sistem kendali linear hanya mampu menyeimbangkan sistem dalam *workspace* yang sempit. Sedangkan kendali nonlinear yang diterapkan pada sistem akan memberikan performa keseimbangan yang lebih baik. Penelitian selanjutnya yang dijadikan rujukan penulis adalah milik *Harun Yetkin* [9]. *Harun Yetkin* dalam jurnal yang diterbitkan oleh *OHIO university* tersebut melakukan penelitian untuk menyeimbangkan sebuah sepeda menggunakan *single gyroscope*. Dalam penelitian tersebut digunakan sistem kendali nonlinear yaitu *sliding mode control*. Sepeda dimodelkan menjadi bentuk *nonlinear state space*. Hasil yang didapatkan yaitu sepeda dapat stabil dengan osilasi sudut *roll* antara dua hingga tiga derajat.

2.2 Giroskop

Giroskop adalah alat berupa roto cakram yang berputar dengan kecepatan tinggi pada sumbunya dan tetap dalam posisinya apabila tidak ada pengaruh kekuatan luar. Giroskop dapat mempertahankan orientasi berlandaskan pada prinsip-prinsip momentum sudut yang diakibatkan oleh putaran poros giroskop. Stabilitas dari giroskop dalam mempertahankan kedudukan akan meningkat jika rotor memiliki momen inersia dan kecepatan putar yang lebih besar. Karakteristik lain dari giroskop adalah precesi. Precesi adalah gerakan memiringkan atau berputar terhadap sumbu giroskop sebagai akibat dari gaya eksternal yang

diterapkan. Presesi giroskop akibat pengaruh gaya eksternal misalnya berupa gravitasi.

2.3 Control Moment Gyroscope

Control Moment Gyroscope (CMG) adalah sebuah cara untuk menghasilkan torsi yang biasa digunakan untuk mengontrol level pada satelit maupun pesawat luar angkasa. CMG terdiri dari rotor dan gimbal berpenggerak motor. Gimbal ini berfungsi untuk mengatur kemiringan dari momentum sudut rotor. Perubahan momentum sudut menghasilkan torsi giroskopik yang digunakan untuk memutar pesawat atau satelit. Prinsip tersebut akan digunakan pada penelitian ini untuk menjaga kestabilan *inverted pendulum*.

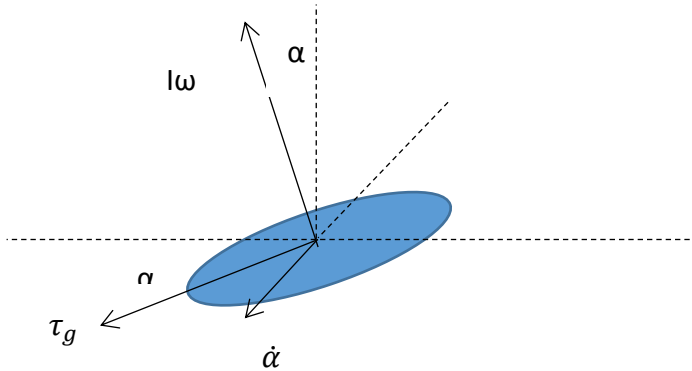
Terdapat dua tipe CMG, *single gimbal* dan *double gimbal*. Pada penelitian ini digunakan tipe *single gimbal* karena lebih sederhana dan memiliki keluaran torsi yang lebih besar dibandingkan dengan tipe *double gimbal*. Selain itu, tipe *single axis* CMG ini memiliki efisiensi energi paling tinggi. Perubahan arah momentum sudut dari rotor menghasilkan giroskopik yang akan bekerja pada *inverted pendulum* yang telah diaplikasikan CMG. Torsi giroskopik yang dihasilkan digunakan untuk menyeimbangkan *inverted pendulum*. Karena memiliki efisiensi yang tinggi, penggunaan *single axis* CMG dapat menghasilkan keluaran torsi yang besar dengan konsumsi listrik yang relatif kecil.

Hukum pertama *Newton* mengatakan bahwa suatu objek yang bergerak akan tetap bergerak kecuali dikenai gaya eksternal. Hukum ini juga berlaku bagi suatu objek yang berputar. Sebuah roda yang berputar dengan kecepatan konstan tanpa dikenai gaya eksternal maka akan tetap berputar pada porosnya. Sedangkan jika sebuah roda berputar yang dikenai gaya dengan resultan tidak sama dengan nol, maka roda tersebut akan mengalami sebuah fenomena yang disebut presesi. Momentum sudut merupakan besaran vektor yang memiliki besar dan arah. Gambar 2.1 menunjukkan arah momentum sudut ($I\omega$), kecepatan putaran gimbal $\dot{\alpha}$, dan torsi giroskopik τ_g . Besar momen inersia flywheel I dan momentum sudut L adalah sebagai berikut:

$$I = \frac{1}{2}mr^2 \quad (2.1)$$

$$L = I\omega \quad (2.2)$$

Parameter m adalah massa *flywheel*, r adalah jari-jari *flywheel*, dan ω adalah kecepatan putar *flywheel*.



Gambar 2.1 Arah momentum sudut ($I\omega$), kecepatan putaran gimbal $\dot{\alpha}$, dan torsi giroskopik τ_g

Untuk mengetahui efek presesi yang disebabkan oleh laju perubahan momentum sudut, *flywheel* harus berputar pada sumbunya dan diberikan putaran pada koordinat gimbal dengan memberi torsi eksternal. Perputaran ini mengakibatkan terjadinya perputaran lain yang secara simultan terjadi pada sumbu yang tegak lurus sumbu putar *flywheel* dan sumbu putar gimbal. Perlu diperhatikan bahwa torsi presesi ini diakibatkan oleh laju perubahan momentum sudut. Dapat dikatakan bahwa torsi presesi hanya akan terjadi pada *flywheel* yang telah diaplikasikan gimbal dengan nilai $|\dot{\alpha}| > 0$, dimana α adalah sudut presesi gimbal dan $\dot{\alpha}$ adalah laju putaran gimbal. Presesi sangat bermanfaat berupa terciptanya suatu torsi yang disebut dengan torsi amplifikasi atau torsi giroskopik. Besar dari torsi presesi atau torsi giroskopik ini bergantung pada nilai kecepatan sudut gimbal dan juga kecepatan sudut *flywheel*. Besarnya output torsi giroskopik τ_g yang dihasilkan, dirumuskan sebagai berikut:

$$\tau_g = \dot{\alpha} \times L \quad (2.3)$$

2.4 Sistem kendali PID

PID (Proportional–Integral–Derivative *controller*) merupakan system kendali yang biasa digunakan pada industri. Karakteristik dari sistem kendali PID adalah adanya umpan balik pada sistem tersebut. PID *Blok Diagram* ditunjukkan pada gambar 2.5.

Komponen kontrol PID ini terdiri dari tiga jenis yaitu Proporsional, Integratif dan Derivatif. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu *plant*. PID adalah salah satu *controller* untuk sistem linear dan juga cukup mudah untuk penggunaannya. Maka dari itu penggunaan PID *controller* dirasa cukup untuk memberikan dampak pada alat penelitian kali ini.

2.4.1 Kontrol Proporsional

K_p adalah konstanta proporsional yang berlaku sebagai *gain* (penguat) saja tanpa memberikan efek dinamik kepada kinerja sistem. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini cukup mampu untuk memperbaiki *transient response* khususnya *rise time* dan *settling time*. Kontrol proporsional memiliki keluaran yang sebanding/proporsional dengan besarnya sinyal error (selisih antara *setpoint* dengan harga aktualnya).

2.4.2 Kontrol Integratif

Kontrol Integral berfungsi untuk mengurangi atau menghilangkan *steady state error*. Jika sebuah *controller* tidak memiliki unsur integrator, control proporsional tidak mampu menjamin keluaran sistem dengan *steady state error*-nya nol.

Kontrol I dapat memperbaiki sekaligus menghilangkan *steady-state error*, namun pemilihan K_i yang tidak tepat dapat menyebabkan *transient response* yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan K_i yang sangat tinggi justru dapat menyebabkan output berosilasi. Keluaran pengontrol ini merupakan hasil penjumlahan yang terus menerus dari error yang terukur. Jika sinyal error tidak mengalami perubahan, maka keluaran akan menjaga keadaan tetap.

2.4.3 Kontrol Derivatif

Keluaran kontrol diferensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan pengontrol

akan mengakibatkan perubahan yang sangat besar dan cepat. Ketika masukannya tidak mengalami perubahan. Dengan sifat ini ia dapat digunakan untuk memperbaiki *transient response* dengan memprediksi error yang akan terjadi. Kontrol Derivative hanya berubah saat ada perubahan error sehingga saat error statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler derivative tidak dapat dipakai sendiri.

Halaman ini sengaja dikosongkan

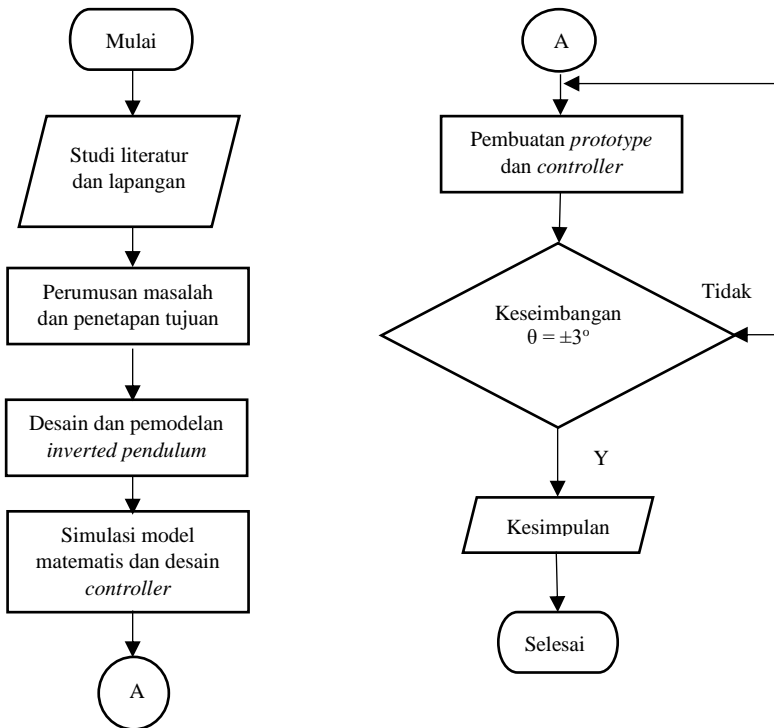
BAB III

METODOLOGI

Metodologi penelitian yang akan digunakan adalah dengan metode simulasi software untuk mendapatkan nilai kontrol PID dan metode pengujian secara *real time* pada *prototype*.

3.1 Diagram Alir Penelitian Penelitian

Penulisan tugas akhir dilakukan untuk menganalisa CMG untuk menyeimbangkan *inverted pendulum*. Dalam menganalisa *inverted pendulum* serta pembuatan sistem kendali PID dilakukan beberapa tahapan yang ditunjukkan oleh diagram alir penelitian pada gambar 3.1 sebagai berikut:



Gambar 3.1 Diagram alir penelitian

3.2 Langkah–langkah Penelitian

Langkah–langkah pada penelitian ini adalah sebagai berikut:

1. Studi Literatur dan Lapangan

Pada tahap studi literatur dilakukan kajian secara teoritis mengenai pemahaman dasar mengenai giroskop, sistem instrumen, rangkaian elektronika dan *software* yang mendukung untuk penyelesaian permasalahan dalam penelitian ini. Studi pustaka yang digunakan diperoleh dari buku, jurnal, internet atau penelitian sebelumnya. Sedangkan studi lapangan dilakukan dengan melakukan kajian terhadap komponen mekanik maupun komponen elektronik yang digunakan pada penelitian ini.

2. Perumusan Masalah dan Penetapan Tujuan Penelitian

Perumusan masalah dalam penelitian ini untuk mengidentifikasi permasalahan guna menjawab latar belakang dari penelitian ini. Perumusan masalah penelitian ini adalah bagaimana cara menyeimbangkan *inverted pendulum* dengan menggunakan penyeimbang giroskop serta bagaimana cara membuat sistem kendali penyeimbang *inverted pendulum*. Sedangkan tujuan penelitian ditetapkan untuk menjawab permasalahan dari penelitian ini, yaitu untuk membuat desain mekanik dan desain sistem kendali untuk menyeimbangkan *inverted pendulum*.

3. Desain dan Pemodelan *Inverted Pendulum*

Setelah melakukan studi literatur dan lapangan, penelitian dilanjutkan dengan mendesain *inverted pendulum*. *Inverted pendulum* yang merupakan sistem mekanik lalu dimodelkan kedalam bentuk matematis. Pemodelan dilakukan agar sistem dapat dapat disimulasikan menggunakan *software* untuk selanjutnya didesain sistem kontrol untuk menyeimbangkan *inverted pendulum* tersebut. Terdapat dua pemodelan yang dilakukan yaitu pemodelan sistem mekanis *inverted pendulum* dan pemodelan aktuator motor servo menggunakan sistem identifikasi.

4. Simulasi Model Matematis dan Desain *Controller*

Setelah didapatkan model matematis dari *inverted pendulum* dan sistem identifikasi motor servo, maka dilakukan desain *controller* yang pada penelitian ini menggunakan sistem kendali PID. Pada tahapan ini, digunakan *software Matlab* untuk membantu mendesain sistem kendali

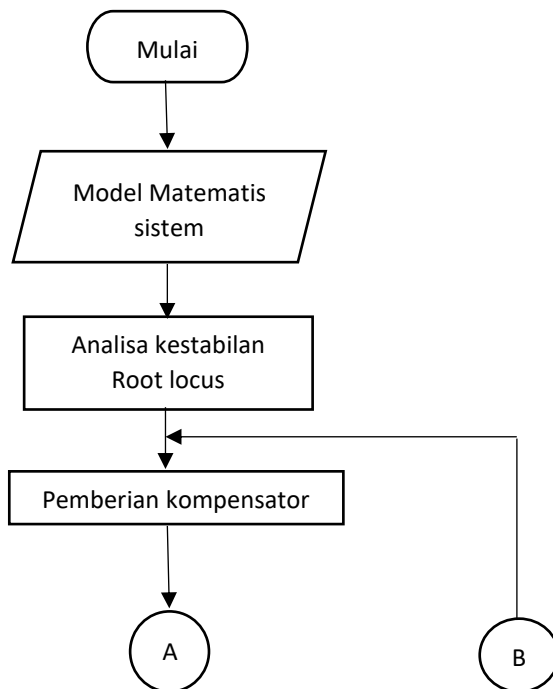
yang paling optimal untuk menyeimbangkan *inverted pendulum*. Diagram alir dari desain *controller* PID ditunjukkan pada gambar 3.2.

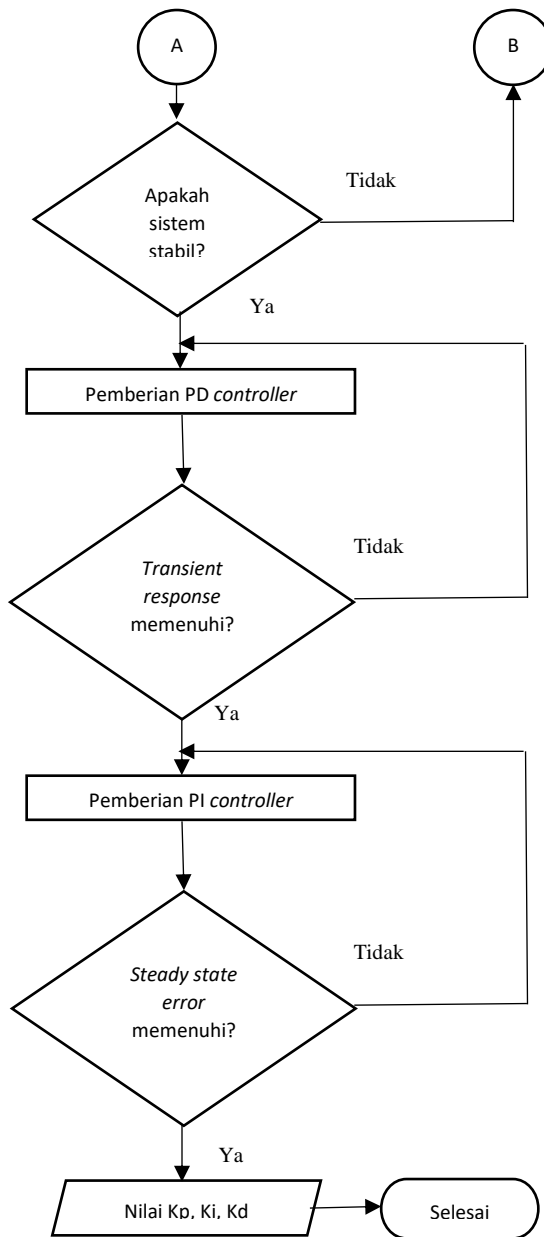
5. Pembuatan *Prototype* dan *Controller*

Dalam perancangan ini, penulis menentukan komponen-komponen yang akan digunakan sesuai kebutuhan dan spesifikasi. Diagram alir yang nantinya akan dijadikan acuan pembuatan *prototype* dan *controller* masing-masing ditunjukkan gambar 3.3 dan gambar 3.4. Sedangkan skema elektronik ditunjukkan pada gambar 3.5 dan 3.6.

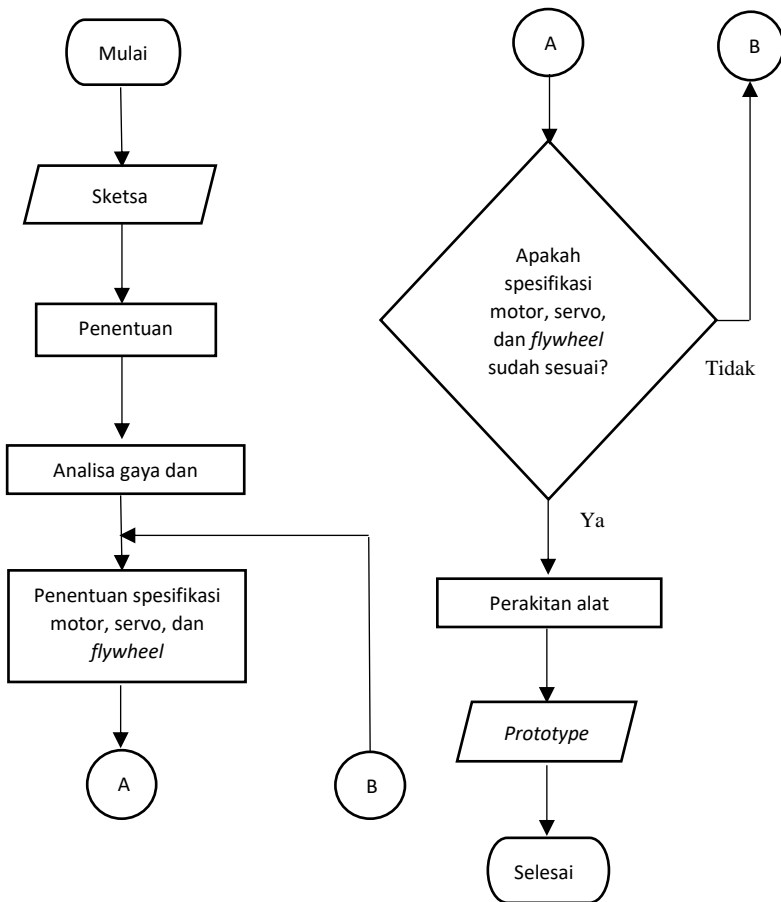
6. Kesimpulan

Kesimpulan diambil dari hasil penelitian yang dilakukan berupa simulasi dan pengujian *prototype*. Kesimpulan juga berupa jawaban dari perumusan masalah dan juga tujuan dari penelitian ini.

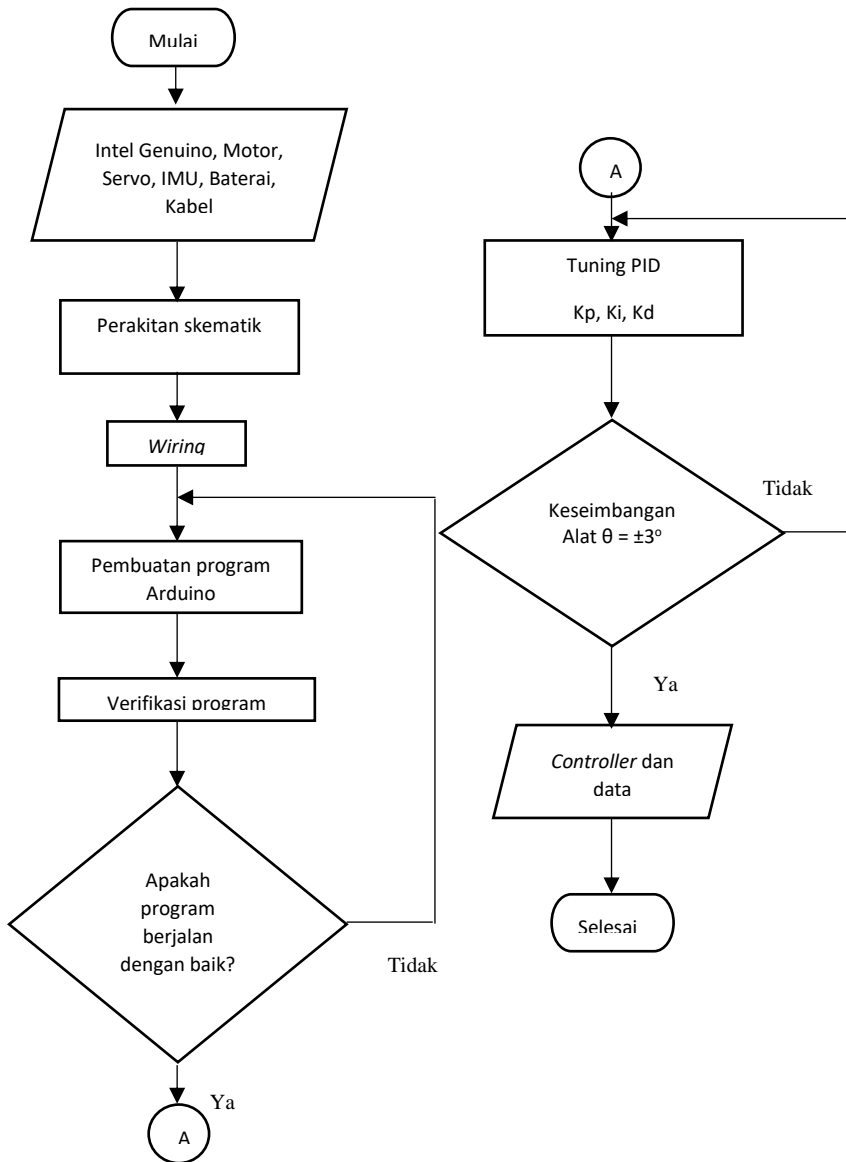




Gambar 3.2 Diagram alir desain PID *controller*



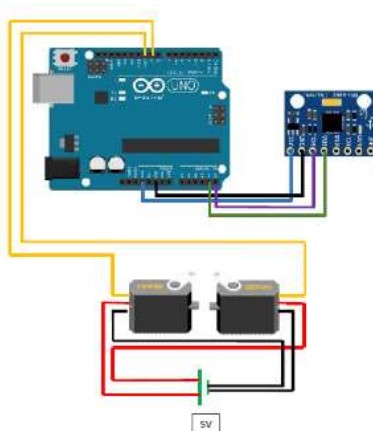
Gambar 3.3 Diagram alir perancangan dan pembuatan *prototype*



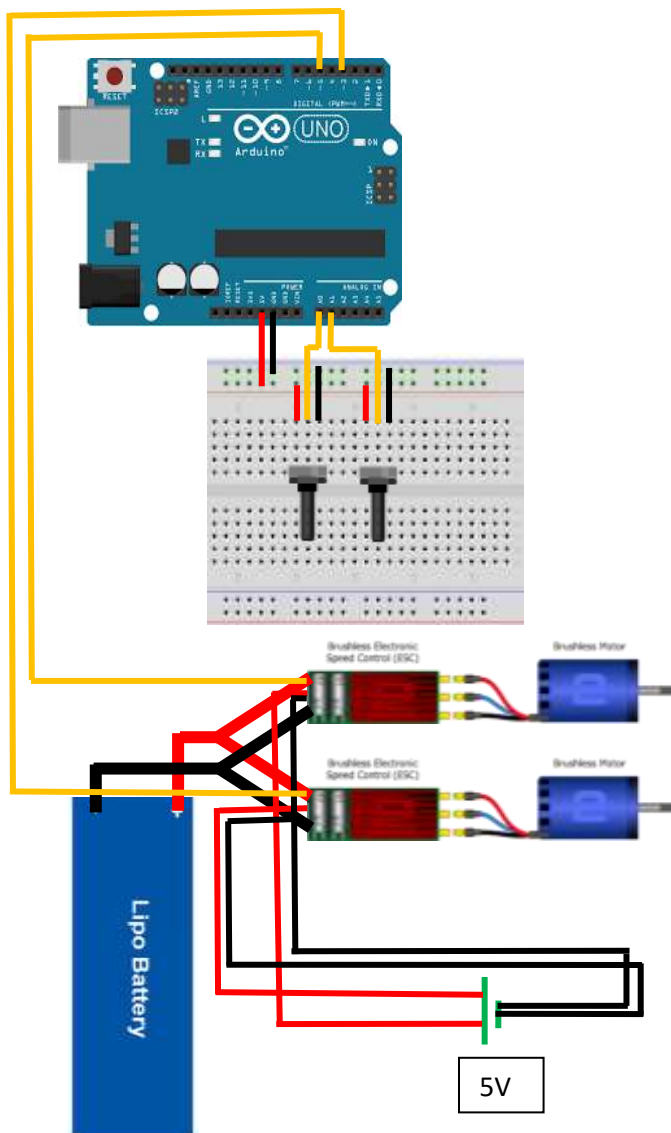
Gambar 3.4 Diagram alir pembuatan *controller*

Tabel 3.1 Komponen mekanik dan elektronik penelitian

| No | Nama Komponen | Jumlah | Satuan |
|----|------------------------------------|--------|-------------|
| 1 | <i>Flywheel</i> | 2 | 0,537 kg |
| 2 | Badan Alat | 1 | 1,09 kg |
| 3 | <i>Motor brushless DC</i> | 2 | 1350kv |
| 4 | <i>Electronic speed controller</i> | 2 | 80A dan 70A |
| 5 | Motor Servo | 2 | 9 kg.cm |
| 6 | Arduino Uno | 2 | - |
| 7 | Sensor MPU6050 | 1 | 6 dof |
| 8 | Baterai LiPo | 1 | 3000 mAh 3s |
| 9 | <i>Potentiometer</i> | 2 | 10K |
| 10 | <i>Power Supply</i> | 1 | 24V 5A |
| 11 | PC | 1 | - |
| 12 | <i>Project Board</i> | 1 | - |



Gambar 3.5 Skema elektronik pengendalian motor servo



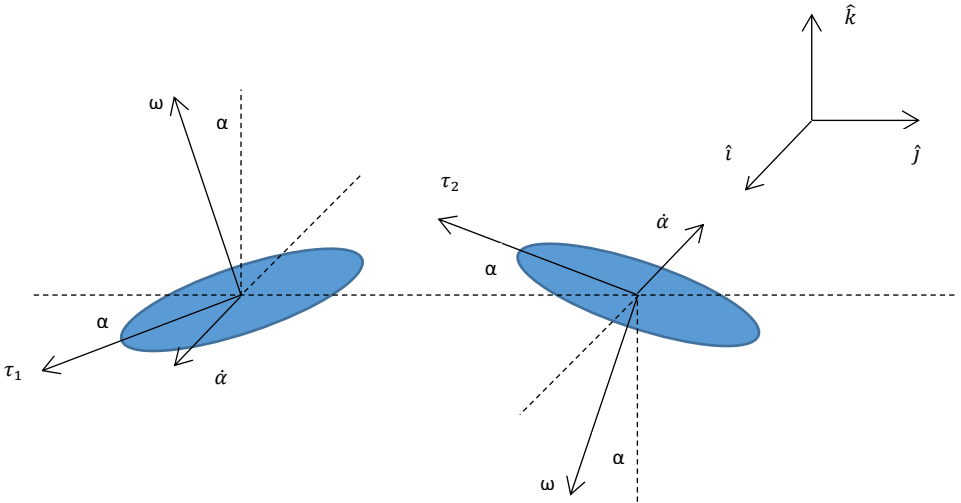
Gambar 3.6 Skema elektronik pengendalian motor brushless DC

BAB IV

ANALISA DAN PEMBAHASAN

4.1 Pemodelan dan Penentuan Spesifikasi Alat

4.1.1 Konfigurasi penyeimbang giroskop



Gambar 4.1 Konfigurasi *flywheel* pada *inverted pendulum* dengan *double gyroscope*

$$\tau_i = \dot{\alpha} \times (I\omega) \quad (4.1)$$

Penelitian ini menggunakan dua giroskop yang diaplikasikan pada *inverted pendulum* yang arah putaran tiap *flywheel* saling berlawanan. Dengan $\dot{\alpha}$ adalah kecepatan sudut gimbal, I momen inersia *flywheel* dan ω adalah kecepatan putar *flywheel*, maka besarnya torsi giroskopik total merupakan resultan torsi giroskopik dari *flywheel* 1 dan torsi giroskopik dari *flywheel* 2. Besarnya torsi giroskopik terhadap variasi kecepatan sudut *flywheel* ditunjukkan pada table 4.1.

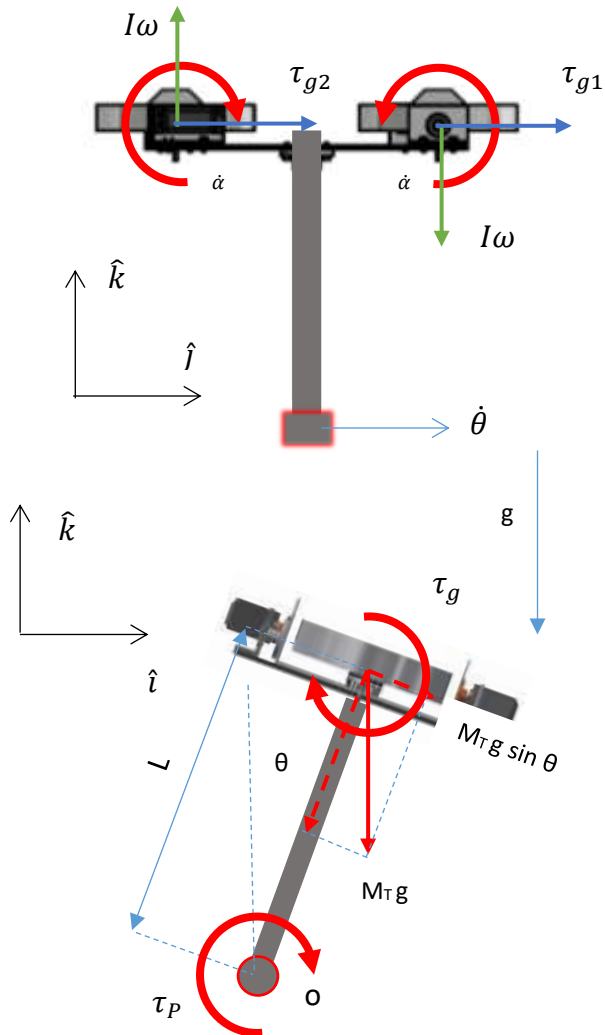
$$\tau_g = \tau_1 + \tau_2 = 2(I\omega) \dot{\alpha} \cos \alpha (-\hat{j}) \quad (4.2)$$

Tabel 4.1 Torsi giroskopik τ_g terhadap variasi kecepatan *flywheel* ω

| ω (rpm) | α dot maks (rpm) | I (kg.m ²) | τ_g maks (Nm) |
|----------------|-------------------------|------------------------|--------------------|
| 1000 | 66 | 0.0009666 | 1.413 |
| 2000 | 66 | 0.0009666 | 2.826 |
| 3000 | 66 | 0.0009666 | 4.239 |
| 4000 | 66 | 0.0009666 | 5.652 |
| 5000 | 66 | 0.0009666 | 7.065 |
| 6000 | 66 | 0.0009666 | 8.478 |
| 7000 | 66 | 0.0009666 | 9.891 |
| 8000 | 66 | 0.0009666 | 11.304 |
| 9000 | 66 | 0.0009666 | 12.718 |
| 10000 | 66 | 0.0009666 | 14.131 |
| 11000 | 66 | 0.0009666 | 15.544 |
| 12000 | 66 | 0.0009666 | 16.957 |
| 13000 | 66 | 0.0009666 | 18.370 |
| 14000 | 66 | 0.0009666 | 19.783 |
| 15000 | 66 | 0.0009666 | 21.196 |
| 16000 | 66 | 0.0009666 | 22.609 |

Gambar 4.2 menunjukkan gaya-gaya yang bekerja pada *inverted pendulum*. Momen gaya *inverted pendulum* akan dilawan olehmomen gaya atau torsi giroskopik untuk menyeimbangkan sistem. Besarnya torsi *inverted pendulum* terhadap variasi sudut ditunjukkan pada table 4.2.

4.1.2 Free body diagram inverted pendulum



Gambar 4.2 Gaya-gaya yang bekerja pada *inverted pendulum*

Equation of motion:

$$\sum M = I_p \ddot{\theta}$$

$$2(I\omega) \dot{\alpha} \cos \alpha + M_T g \sin \theta L = I_p \ddot{\theta}$$

$$2(I\omega) \dot{\alpha} \cos \alpha + M_T g \sin \theta L = (M_T L^2) \ddot{\theta}$$

$$\ddot{\theta} = \frac{g}{L} \sin \theta + \frac{2(I\omega)}{M_T L^2} \dot{\alpha} \cos \alpha$$

Linearisasi persamaan gerak *inverted pendulum* pada $\theta \approx 0$ dan $\alpha \approx 0$, sehingga:

$$\ddot{\theta} = \frac{g}{L} \theta + \frac{2(I\omega)}{M_T L^2} \dot{\alpha}$$

$$\ddot{\theta} - \frac{g}{L} \theta = \frac{2(I\omega)}{M_T L^2} \dot{\alpha}$$

Transformasi *laplace* untuk persamaan gerak *inverted pendulum* menjadi:

$$s^2 \theta - \frac{g}{L} \theta = \frac{2(I\omega)}{M_T L^2} \dot{\alpha}$$

$$\theta(s^2 - \frac{g}{L}) = \frac{2(I\omega)}{M_T L^2} \dot{\alpha}$$

$$\frac{\theta}{\dot{\alpha}} = \frac{\frac{2(I\omega)}{M_T L^2}}{s^2 - \frac{g}{L}} \quad (4.3)$$

Input variable $C(s)$: $\dot{\alpha}$ (kecepatan sudut gimbal)

Output variable $R(s)$: θ (sudut roll)

4.1.3 Penentuan Spesifikasi Elektronik

Metodologi penentuan spesifikasi alat merujuk pada diagram alir 3.3 yaitu diawali dengan menentukan dimensi alat serta analisa gaya dan momen. Hal tersebut bertujuan agar alat dapat berjalan dengan baik dan bekerja sebagaimana mestinya. Pada penelitian ini terdapat beberapa

Tabel 4.2 Torsi *inverted pendulum* τ_p terhadap variasi sudut kemiringan θ

| θ (deg) | Mt(kg) | g (m/s ²) | L (m) | τ_p (Nm) |
|----------------|--------|-----------------------|-------|---------------|
| 5 | 2.614 | 9.81 | 0.2 | 0.447 |
| 10 | 2.614 | 9.81 | 0.2 | 0.891 |
| 15 | 2.614 | 9.81 | 0.2 | 1.327 |
| 20 | 2.614 | 9.81 | 0.2 | 1.754 |
| 25 | 2.614 | 9.81 | 0.2 | 2.167 |
| 30 | 2.614 | 9.81 | 0.2 | 2.564 |
| 35 | 2.614 | 9.81 | 0.2 | 2.942 |
| 40 | 2.614 | 9.81 | 0.2 | 3.297 |
| 45 | 2.614 | 9.81 | 0.2 | 3.627 |

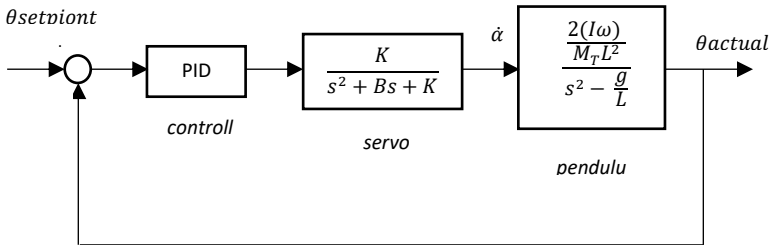
variabel tetap yang nilainya tidak dapat divariasikan diantaranya adalah massa, gaya gravitasi, dan momen inersia dari *flywheel*. Sedangkan kecepatan putar *flywheel* dapat divariasikan dengan nilai maksimal pada 16000 rpm tetapi tidak masuk sebagai variabel kontrol (*controlled variable*). Kecepatan putar servo (gimbal) merupakan variabel kontrol (*controlled variable*) pada penelitian ini. Nilai Kecepatan putar servo ditentukan oleh controller dengan nilai maksimum sebesar 66 rpm.

Pada penelitian ini ditentukan untuk kecepatan putar *flywheel* sebesar 5000 rpm. Secara teori, berdasarkan perhitungan yang ditunjukkan pada table 4.1, besar torsi giroskopik maksimal yang tercipta adalah sebesar 7 Nm. Torsi tersebut sangat mencukupi untuk melawan torsi bodi pendulum pada sudut 45°, dengan catatan *controller* memberikan kecepatan maksimum pada servo. Dalam penentuan spesifikasi motor *brushless* DC, motor *brushless* DC diharuskan dapat memutar *flywheel* seberat 537 gram dengan momen inersia sebesar 0.0009666 kg.m². Kendala pada penentuan spesifikasi motor *brushless* DC pada penelitian ini adalah tidak terdapat spesifikasi untuk torsi di pasaran, sehingga penulis mencoba beberapa jenis motor *brushless* DC berdasarkan diameter motor. Dari hasil percobaan, *flywheel* dapat

berputar dengan baik menggunakan motor berdiameter 35 mm. Spesifikasi motor *brushless* yang digunakan adalah 1350kv, diameter dan tebal 35-30, dan menggunakan *electronic speed controller* dengan spesifikasi arus maksimum 80A.

Penentuan spesifikasi servo berdasarkan pada torsi yang dihasilkan servo. Massa total dari gimbale, motor *brushless*, dan flywheel pada percobaan kali ini adalah sebesar 700 gram. Agar dapat berjalan dengan baik, servo harus dapat memutar gimbale dengan beban 700 gram. Pada penelitian ini digunakan servo dengan spesifikasi 9,4 kg.cm yang dirasa sangat cukup untuk memutar beban.

4.2. Blok Diagram Pengendalian



Gambar 4.3 Blok diagram pengendalian keseimbangan *inverted pendulum*

Controlled variable: $\dot{\alpha}$ (kecepatan sudut gimbale)

Manipulated variable: PWM Servo

Pada penelitian ini digunakan sistem kendali PID untuk menyeimbangkan alat yakni menyeimbangkan sudut θ pada *set point* kemiringan 0° . Sistem mendapatkan input berupa kecepatan sudut gimbale, dan output berupa sudut kemiringan θ . Aktuator pada penelitian ini berupa servo yang akan dilakukan system identifikasi pada subbab berikutnya, sedangkan *feedback* yang diberikan berupa sudut kemiringan θ yang diukur oleh sensor MPU6050. Blok diagram pengendalian sistem *inverted pendulum* ditunjukkan pada gambar 4.3

Sistem disimulasikan menggunakan *Matlab Simulink* untuk mendapatkan nilai kontrol PID yang optimal. Selanjutnya nilai PID dari hasil simulasi akan diterapkan pada alat di eksperimen. Nilai-nilai

parameter percobaan yang digunakan pada penelitian kali ini adalah sebagai berikut:

Tabel 4.3 Nilai parameter percobaan

| Parameter | Nilai | Satuan |
|-----------------------|----------|-------------------|
| Mf1 | 0.537 | kg |
| Mf2 | 0.537 | kg |
| Rf | 0.06 | m |
| I | 0.000967 | kg.m ² |
| ω @5000rpm | 523 | rad/s |
| $\dot{\alpha}$ (maks) | 6.98 | rad/s |
| Mt | 2.614 | kg |
| g | 9.81 | m/s ² |

4.3 Complementary Filter

Sistem kendali PID memerlukan *feedback* untuk menentukan nilai error terhadap *setpoint*. Seperti yang telah disebutkan pada subbab 4.2 bahwa sensor yang digunakan untuk memberikan feedback pada sistem berupa sudut θ terukur adalah sensor MPU6050. Sensor MPU6050 merupakan salah satu sensor IMU(*Inertia Measurement Unit*) yang memiliki 6 output nilai keluaran. Enam nilai output tersebut yakni tiga *axis accelerometer* dan tiga *axis gyroscope* masing-masing berturut-turut a_x , a_y , a_z , dan g_x , g_y , g_z .

4.3.1 Masalah pada *accelerometer*

Accelerometer dapat mengukur semua gaya yang bekerja pada suatu objek. Akan terdapat banyak gaya-gaya yang dapat terukur tidak hanya gravitasi saja. Setiap gaya-gaya kecil sekalipun yang bekerja pada objek akan mengganggu pengukuran. Hal ini yang membuat nilai pengukuran menjadi sangat acak apabila terkena gaya dari luar meskipun kecil, sebagai contoh adalah getaran.

4.3.2 Masalah pada gyroscope

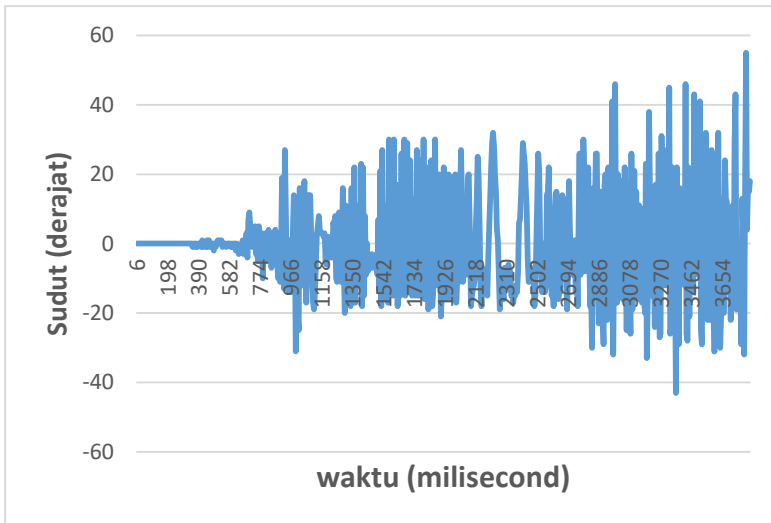
Gyroscope dapat memberikan pengukuran yang akurat yang tidak terpengaruh oleh gaya dari luar. Hal yang menjadi kelemahan dari *gyroscope* adalah nilai pengukuran yang didapat tidak bias kembali meskipun sistem telah kembali ke posisi semula.

4.3.3 Filter

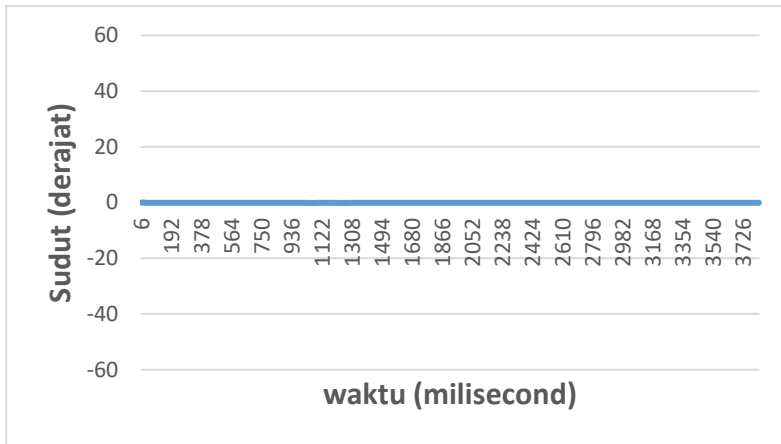
Penelitian ini menggunakan *complementary filter* untuk menyaring data yang diberikan oleh sensor IMU. Prinsip dari *complementary filter* yaitu menggabungkan dua kemampuan dari dari *accelerometer* dan *gyroscope* untuk mendapatkan nilai pengukuran yang baik *complementary filter* dirumuskan sebagai berikut:

$$\text{sudut} = 0,98 * (\text{sudut} + \text{gyrData} * \text{dt}) + 0,02 * \text{accData} \quad (4.4)$$

Fungsi dari *complementary filter* adalah mengambil nilai sudut baru dari *gyroscope* setiap waktu. *Filter* lalu mengecek nilai gaya-gaya



Gambar 4.4 Data sensor IMU tanpa filter



Gambar 4.5 Data sensor IMU menggunakan *Complementary filter*

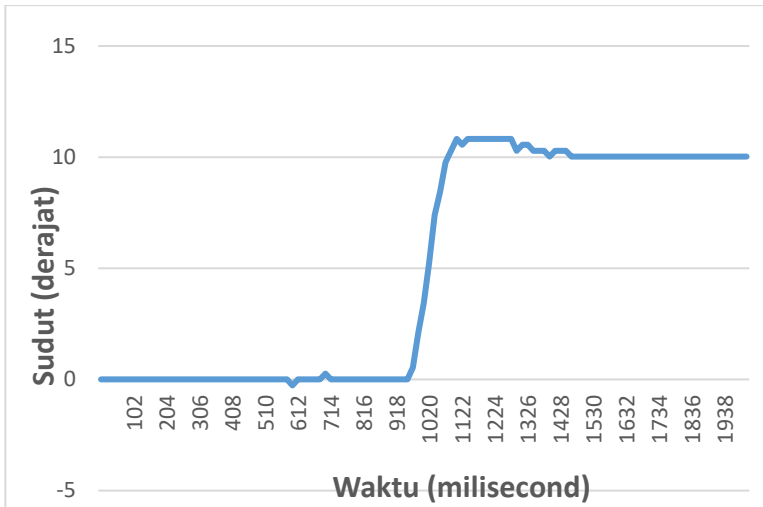
yang terukur dari *accelerometer*. Jika nilai dari gaya-gaya tersebut terlalu besar atau terlalu kecil, *filter* menganggapnya sebagai gangguan dan tidak akan memasukkan nilai kedalam perhitungan. Nilai pengukuran tanpa menggunakan *complementary filter* ditunjukkan pada gambar 4.4, sedangkan nilai pengukuran dengan menggunakan *complementary filter* ditunjukkan pada gambar 4.5

4.4 Sistem Identifikasi Servo

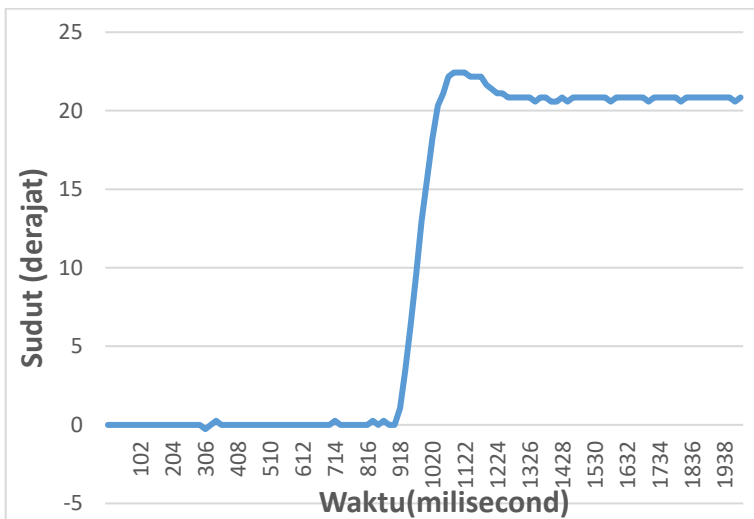
Untuk mendapatkan model matematis dari servo diperlukan sistem identifikasi. Pada gambar 4.3, servo ditunjukkan memiliki model matematis dalam bentuk *second order*. Namun, nilai dari K dan B belum diketahui. Pada subbab ini akan dijelaskan sistem identifikasi untuk servo yang digunakan pada penelitian ini.

Metodologi Sistem Identifikasi Servo

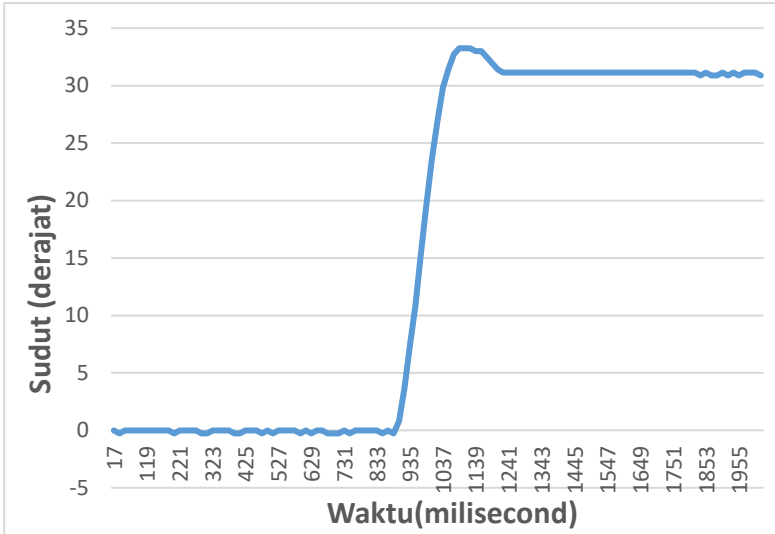
Metodologi yang digunakan untuk mengidentifikasi servo yakni dengan mengukur respon yang diberikan pada servo. Input yang diberikan adalah *step input* berupa sudut tertentu. Sudut-sudut yang diberikan merupakan sudut perkiraan dimana servo bekerja yakni pada sudut 10°, 20°, 30°. Respon dari *step input* untuk sudut 10°, 20°, 30° masing-masing diberikan pada gambar 4.6 , 4.7 , 4.8.



Gambar 4.6 *Step response* motor servo pada setpoint 10°



Gambar 4.7 *Step response* motor servo pada setpoint 20°



Gambar 4.8 *Step response* motor servo pada setpoint 30°

Dari ketiga grafik *step response* servo tersebut dapat disimpulkan bahwa servo merupakan *underdamped system* yang dapat dimodelkan sebagai *second order system*. Ketiga grafik memiliki overshoot terhadap *set point* sebelum mencapai *steady state*. Khusus untuk *set point* sudut 10°, grafik menunjukkan bahwa sistem memiliki nilai *steady state error* sedangkan dua grafik lainnya memiliki nilai *steady state error*.

Second order approximation dari motor servo dirumuskan sebagai berikut:

$$G(S) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (4.5)$$

$$G(S) = \frac{K}{s^2 + Bs + K} \quad (4.6)$$

Untuk mengevaluasi nilai K dan B, dibutuhkan nilai parameter *Peak Time Tp* dari *percent overshoot (%OS)* motor servo. Nilai parameter *percent overshoot %OS* dan *Peak Time Tp* untuk masing-masing *setpoint*

ditunjukkan pada table 4.3. Nilai damping ratio ζ dan frekuensi natural ω_n didapatkan dari persamaan 4.7 dan 4.8.

Tabel 4.4 Nilai Parameter *Transient Response*

| <i>Set Point</i> | <i>%OS</i> | <i>Peak Time(Tp)</i> |
|------------------|------------|----------------------|
| 10 | 7.876 % | 153 ms |
| 20 | 7.578 % | 187 ms |
| 30 | 6.776 % | 204 ms |

$$Tp = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} \quad (4.7)$$

$$\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 - \ln^2(\%OS/100)}} \quad (4.8)$$

Tabel 4.5 Nilai ω_n dan ζ untuk tiap-tiap setpoint

| <i>Setpoint</i> | ω_n | ζ |
|-----------------|------------|---------|
| 10° | 33.970 | 0.629 |
| 20° | 28.130 | 0.635 |
| 30° | 26.706 | 0.650 |
| Rata-Rata | 29.602 | 0.638 |

Maka, *Second order approximation* dari motor servo adalah sebagai berikut:

$$G_{servo}(s) = \frac{K}{s^2 + Bs + K}$$

$$K_{value} = 876.295$$

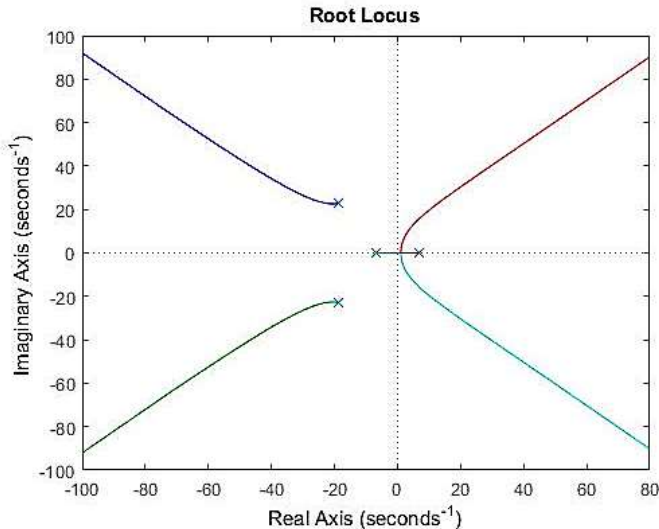
$$B_{value} = 37.774$$

4.5 Analisa Stabilitas

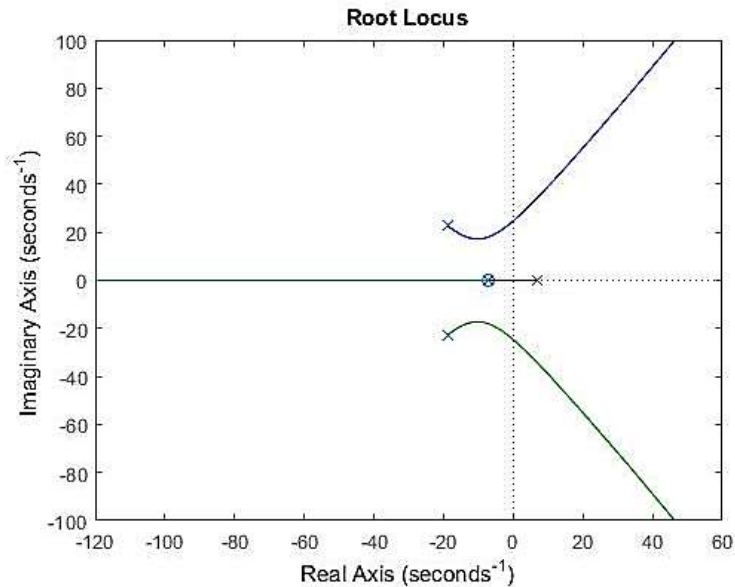
Pada penelitian ini digunakan analisa kestabilan menggunakan metode *root locus*. Metode *root locus* digunakan untuk mengetahui letak dari *pole* pada persamaan karakteristik dari sistem *inverted pendulum* berdasarkan perubahan *gain*-nya. Kestabilan dicapai ketika semua *pole* berada pada *Left Half Plane* (LHP).

Pada gambar 4.9, terdapat satu *pole* yang berada pada *right half plane* (RHP). Hal tersebut mengakibatkan sistem menjadi tidak stabil untuk semua nilai *gain* K. Untuk menstabilkan system, diberikan sebuah *zero* pada $s=-7,0036$ yang secara efektif membatalkan atau menghilangkan satu *pole* pada *real axis*. Efek dari penambahan sebuah *zero* tersebut yakni *gain* sistem dapat disesuaikan agar semua *pole* berada pada LHP agar sistem mencapai kestabilan. Gambar 4.10 menunjukkan efek dari penambahan *zero* pada $s = -7,0036$. Untuk nilai *gain* tertentu, sistem akan mencapai kestabilan. *Controller* dari penambahan *zero* tersebut yang berbentuk PD *controller* dapat dituliskan sebagai berikut:

$$Z(s) = K(s+7,0036) \quad (4.9)$$



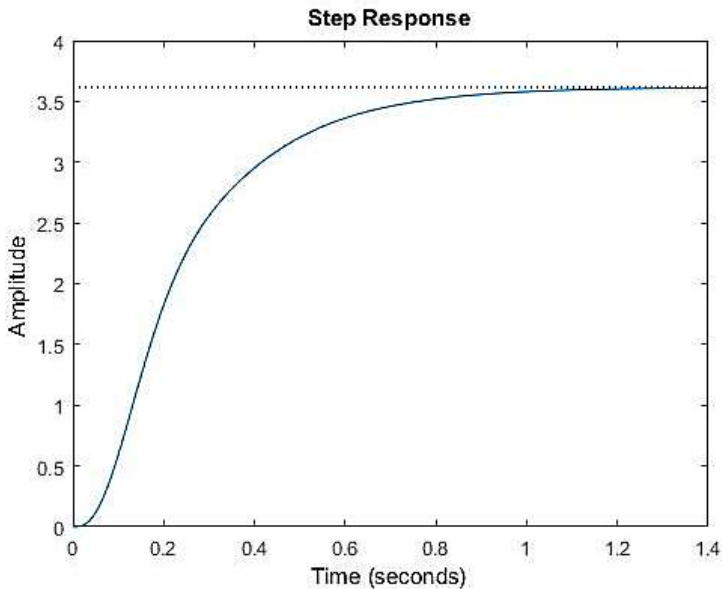
Gambar 4.9 *Root locus* dari sistem *inverted pendulum*



Gambar 4.10 *Root locus* dari sistem *inverted pendulum* dengan kompensasi zero pada $s = -7,0036$

4.6 Desain Sistem Kendali PID

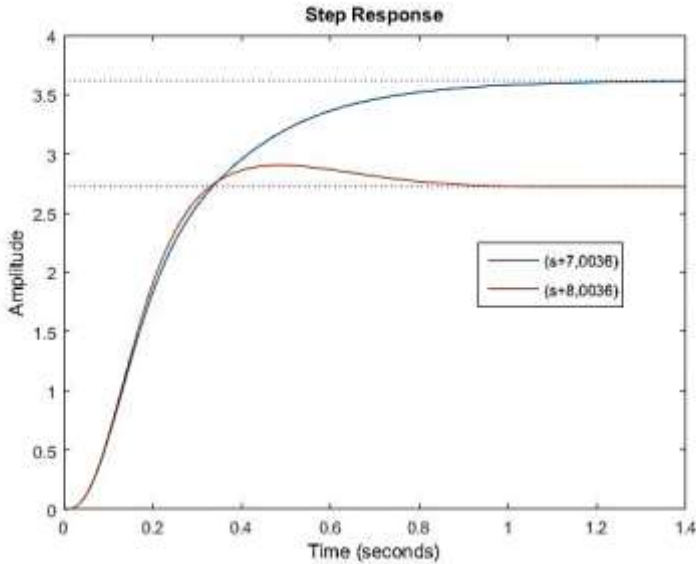
Pada subbab 4.5, telah didapatkan kestabilan dengan menambahkan *zero* yang merupakan sebuah *controller* PD. Penelitian ini menggunakan metoologi desainsistem kendali PID mengikuti diagram alir 3.2. Tahap pertama yang dilakukan dengan memberikan kendali PD pada sistem untuk mendapatkan *transient response* yang diinginkan dari sistem. Pada analisa kestabilan telah didapatkan kompensasi untuk menjadikan sistem untuk mencapai kestabilan. Kompensator tersebut adalah merupakan suatu kendali PD. Kendali PD yang didapatkan pada analisa kestabilan akan dievaluasi lagi agar mendapatkan *transient response* yang diinginkan. Untuk mengetahui *transient response* dari suatu sistem maka kita memberikan *step input* kepada sistem tersebut. *Transient response* untuk PD controller $Z(s) = K(s+7,0036)$ ditunjukkan pada gambar 4.11.



Gambar 4.11 *Step response inverted pendulum*

Tahap selanjutnya yakni mengevaluasi sistem dengan mendesain kembali kendali PD. Untuk meningkatkan *transient response* dari sistem *iverted pendulum*, maka kendali PD diberikan dengan nilai kendali $D(s) = K(s+8,0036)$. Kendali PD tersebut memberikan *transient response* yang lebih baik dibandingkan dengan kompensator PD yang didapatkan dari analisa kestabilan pada subbab 4.5. Gambar 4.12 adalah perbandingan dari dua nilai kendali PD.

Tahap selanjutnya yaitu memberikan sistem kendali PI untuk mencapai *steady state error* yang diinginkan. Pada penelitian ini, penulis berusaha mencapai nilai *steady state error* nol atau menghilangkan *steady state error*. Kendali PI diberikan dengain nilai kendali $I(s) = \frac{s+0,8}{s}$. Sistem juga diberikan *gain* K sebesar 1.2. Dengan pemberian kendali PI ini, maka secara keseluruhan sistem mendapatkan kendali PID dengan nilai kendali dirumuskan pada persamaan 4.10 dan 4.11.



Gambar 4.12 *Step response* dari dua nilai kendali PD

$$PID(s) = \frac{K(s+8,0036)(s+0,8)}{s} \quad (4.10)$$

$$PID(s) = \frac{1,2(s+8,0036)(s+0,8)}{s} = \frac{1,2(s^2+8,8s+6,4)}{s} \quad (4.11)$$

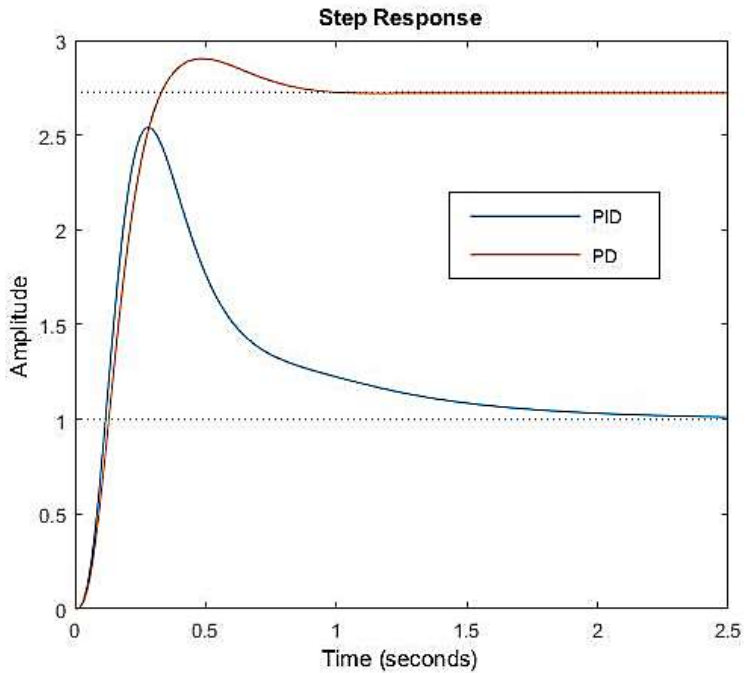
Pada gambar 4.13, ditunjukkan bahwa pengaruh kontrol PI terhadap sistem yaitu membuat nilai *steady state error* sistem menjadi 0. Untuk mendapatkan nilai konstanta k_p , k_i , dan k_d , persamaan sistem kendali PID dievaluasi menggunakan persamaan berikut:

$$G_{pid}(s) = k_p + \frac{k_i}{s} + k_d s = \frac{k_d(s^2 + \frac{k_p}{k_d}s + \frac{k_i}{k_d})}{s} \quad (4.12)$$

$$K_d = 1.2$$

$$\frac{k_p}{k_d} = 8,8, \quad k_p = 7,33$$

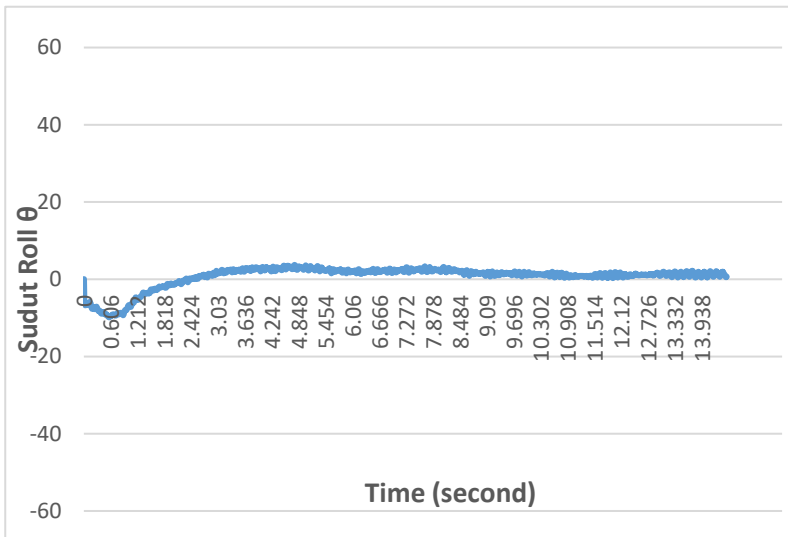
$$\frac{k_i}{k_d} = 6.4, \quad k_i = 5,33$$



Gambar 4.13 Step response sistem *inverted pendulum* dengan kendali PD dan PID

4.7 Eksperimen

Nilai kontrol yang didapatkan dari simulasi menjadi nilai awal parameter kontrol yang akan diimplementasikan pada alat yaitu dengan nilai k_p , k_i , k_d masing-masing 7,33, 5,33, 1,2.. Pada gambar 4.14, ditunjukkan grafik sudut *roll* terhadap waktu. Dapat disimpulkan dari grafik tersebut bahwa sistem *inverted pendulum* dapat mempertahankan keseimbangan selama delapan detik. Dalam percobaan lain, *inverted pendulum* mampu mempertahankan kesetimbangan selama delapan hingga sepuluh detik. Sistem akan mengalami osilasi lalu jatuh setelah lebih dari rentang waktu tersebut.

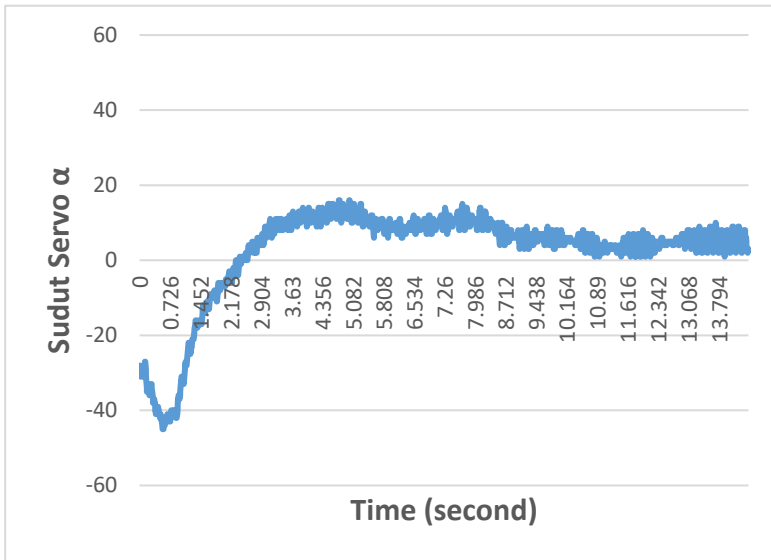


Gambar 4.14 Grafik sudut *roll* θ terhadap waktu

Respon sudut *roll* masih memiliki *error steady state* dari *setpoint* keseimbangan pada 0° , yakni pada sudut sekitar 1° . Sistem *inverted pendulum* cenderung akan jatuh jika telah mengalami osilasi diatas 5° .

Gambar 4.15 menunjukkan grafik sudut servo untuk *flywheel* 1 terhadap waktu. Grafik sudut servo untuk *flywheel* 1 dan *flywheel* 2 adalah identik karena datang dari input yang sama, hanya saja arahnya berbeda atau berlawanan. Perlu digaris bawahi bahwa respon yang didapat pada gambar 4.14 merupakan data yang didapatkan Arduino. Artinya data tersebut merupakan data yang diinginkan atau dikirimkan *controller* kepada kedua servo dan bukan merupakan data sudut servo aktual. Grafik tersebut menunjukkan bahwa pergerakan servo proporsional dibandingkan grafik sudut, dengan sudut maksimal pada 17° pada saat sistem *inverted pendulum* mengalami kesetimbangan.

Secara teoritis, sistem harusnya mampu mempertahankan keseimbangan selama asumsi pada linearisasi masih berlaku yaitu saat $\sin\theta=\theta$ dan $\cos\alpha=1$. Jika nilai-nilai tersebut tidak memenuhi, maka sistem



Gambar 4.15 Grafik sudut α terhadap waktu

kendali yang sudah didesain tidak akan berlaku. Namun, sistem *inverted pendulum* dalam eksperimennya hanya dapat menyeimbangkan diri selama delapan hingga sepuluh detik. Hal ini diakibatkan oleh beberapa sebab diantaranya adalah pengerjaan alat yang tidak presisi. Alat yang tidak presisi mengakibatkan titik berat sistem akan bergeser. Hal tersebut menyebabkan alat yang kita modelkan akan berbeda dengan keadaan actual sehingga parameter kendalinya menjadi tidak maksimal diterapkan pada sistem tersebut. Sebaiknya pengerjaan alat menggunakan mesin cnc dibandingkan menggunakan mesin manual untuk mendapatkan tingkat kepresisian yang tinggi.

Pengerjaan alat yang tidak presisi berimplikasi pada terciptanya vibrasi yang cukup terasa saat alat melakukan *running*. Hal ini juga menjadi salah satu penyebab sistem mengalami osilasi hingga tidak dapat mempertahankan keseimbangan. Untuk penelitian mendatang, dihindari untuk memberikan peredam vibrasi jika memungkinkan.

Pada penelitian ini, digunakan sistem kendali PID yang merupakan sistem kendali linear. Seperti yang telah dijelaskan sebelumnya, linearisasi pada pemodelan sistem *inverted pendulum* yaitu pada sudut θ dan α mendekati 0 sehingga nilai $\sin\theta=\theta$ dan $\cos\alpha=1$. Jika nilai pendekatan tersebut tidak memnuhi maka kendali yang diberikan pada sistem juga tidak lagi berlaku. Untuk mendapatkan performa keseimbangan yang baik, sebaiknya digunakan pengendalian nonlinear. Dengan sistem kendali nonlinear maka parameter kendali akan menyesuaikan kondisi sudut kemiringan, sehingga parameter kendali akan menyesuaikan sendiri secara otomatis.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada penelitian ini didapatkan rancangan sistem *self-balancing* pada *inverted pendulum* dengan berat total sebesar 2,6 kg dan panjang atau tinggi titik berat sebesar 20 cm. Sistem memiliki dua penyeimbang berbasis giroskop dengan massa masing-masing *flywheel* sebesar 537 gram. Sistem *self-balancing* berjalan pada kecepatan *flywheel* sebesar 5000 rpm dengan spesifikasi motor 35-30 1350 kv (1350rpm/V). Sedangkan kecepatan maksimum servo (gimbal) 66 rpm dengan torsi 9,4 kg.cm yang dikendalikan dengan sistem kendali berbasis PID.

Hasil dari penelitian ini adalah didapatkannya sistem kendali PID yang dapat menyeimbangkan sistem *inverted pendulum*. Stabilisasi *inverted pendulum* terjadi dalam kisaran waktu selama delapan hingga sepuluh detik. Setelah selang waktu tersebut, sistem akan mengalami osilasi dan menjadi tidak stabil.

Linierisasi model dinamis dari sistem *inverted pendulum* secara efektif dapat menstabilkan sistem dengan desain sistem kendali yang sederhana. Namun, sistem kendali tersebut hanya akan berlaku pada rentang *space* yang sempit. Lebih dari itu, maka sistem kendali tidak akan bisa menyeimbangkan sistem.

5.2 Saran

Untuk penelitian selanjutnya disarankan:

1. Sebaiknya pembuatan alat menggunakan mesin cnc dengan tingkat kepresisian yang tinggi, karena struktur mekanik yang buruk akan membuat alat susah dikontrol dan tidak sesuai saat dimodelkan.
2. Sebaiknya digunakan sistem kendali non linear untuk penelitian ini jika ingin mendapatkan performa kestabilan yang tinggi. Hal tersebut dikarenakan linearisasi yang dilakukan pada sudut 0° baik untuk sudut roll dan sudut gimbal pada penelitian ini menyebabkan sistem kendali yang telah dirancang hanya berlaku pada sudut mendekati 0° saja.

DAFTAR PUSTAKA

- [1] Elbert Hendricks, O. J. (2008). *Linear Systems Control*. Berlin: Springer.
- [2] Girard, S. C. (2008). Gyroscopic Stabilization of Unstable Vehicles: Configurations, Dynamics, and Control.
- [3] Hibbeler, R. C. (2010). *ENGINEERING MECHANICS DYNAMICS TWELFTH EDITION*. New Jersey: Pearson.
- [4] Lam, P. Y. (2012). DESIGN AND DEVELOPMENT OF A SELF-BALANCING BICYCLE USING CONTROL MOMENT GYRO.
- [5] Nise, N. S. (2013). *Control Systems Engineering 6th Edition*. California: John Wiley & Sons, Inc.
- [6] POLRI, K. (2017, Maret 15). *Jenis Kendaraan*. Retrieved from KORLANTAS POLRI Web site: <http://www.korlantas-irsms.info/graph/vehicleTypeData>
- [7] POLRI, K. (2017, Maret 15). *Jumlah Data Kendaraan Berdasarkan Tahun Pembuatan*. Retrieved from Korp Lalu Lintas POLRI Web site: <http://www.rckorlantaspolri.id/infokendaraan.php>
- [8] Walck, C. J. (2013). Stabilization of an inverted pendulum using control.
- [9] Yetkin, H. (n.d.). Gyroscopic Stabilization of an Unmanned Bicycle.

LAMPIRAN

Program sensor IMU, motor servo, dan PID

```
#include<Wire.h>
#include<Servo.h>
Servo A;
Servo B;
const int MPU = 0x68;
int t = 0, dt = 1;
int AcX, AcY, AcZ, GyX, GyY, GyZ, tmp;
int AcXo, AcYo, AcZo, GyXo, GyYo, GyZo;
int sudut_servo;
float roll = 0, pitch = 0, rollgy = 0, pitchgy = 0, rollac = 0, pitchac = 0, Ax, Ay, Az, Gx, Gy, Gz, gain = 0.95;
float pid_error_temp;
float pid_i_mem_roll, pid_roll_setpoint, gyro_roll_input,
pid_output_roll, pid_last_roll_d_error;
float pid_p_gain_roll = 7.3;           //Gain setting for the roll P-
controller
float pid_i_gain_roll = 5.33;         //Gain setting for the roll I-
controller
float pid_d_gain_roll = 1.2;         //Gain setting for the roll D-
controller
int pid_max_roll = 200;               //Maximum output of the PID-
controller (+/-)
void MPUconfig(int Addr, int data)
{
    Wire.beginTransmission(MPU);
    Wire.write(Addr);
    Wire.write(data);
    Wire.endTransmission();
}
void MPUread()
```

```

{
    Wire.beginTransaction(MPU);
    Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
    Wire.endTransmission();
    Wire.requestFrom(MPU,14); // request a total of 14 registers
    AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) &
0x3C (ACCEL_XOUT_L)
    AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) &
0x3E (ACCEL_YOUT_L)
    AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) &
0x40 (ACCEL_ZOUT_L)
    tmp=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) &
0x3C (ACCEL_XOUT_L)
    GyX=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) &
0x3E (ACCEL_YOUT_L)
    GyY=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) &
0x40 (ACCEL_ZOUT_L)
    GyZ=Wire.read()<<8|Wire.read();
}
void offset()
{
    MPUconfig(0x1A,0b00000000);
    int ax = 0, ay = 0,az = 0,gx = 0,gy = 0,gz = 0,i;
    MPUread();
    MPUread();
    ax = AcX;
    ay = AcY;
    az = AcZ;
    gx = GyX;
    gy = GyY;
    gz = GyZ;
    for(i=0;i<1100;i++)
    {
        MPUread();

```

```

    if(i>100)
    {
        ax = (ax+AcX)/2;
        ay = (ay+AcY)/2;
        az = (az+AcZ)/2;
        gx = (gx+GyX)/2;
        gy = (gy+GyY)/2;
        gz = (gz+GyZ)/2;
    }
    delay(2);
}
AcXo = ax;
AcYo = ay;
AcZo = 16384-az;
GyXo = gx;
GyYo = gy;
GyZo = gz;
MPUconfig(0x1A,0b00000110);
}
void setup()
{
    Serial.begin(115200);
    Wire.begin();
    A.attach(9);
    B.attach(10);
    MPUconfig(0x1A,0b00000110);
    MPUconfig(0x1B,0b00000000);
    MPUconfig(0x1C,0b00000000);
    MPUconfig(0x6B,0b00000000);
    offset();
}
void loop()
{

```

```

t = millis();
MPUread();
Ax = (float)(AcX-AcXo)/16384;
Ay = (float)(AcY-AcYo)/16384;
Az = (float)(AcZ-AcZo)/16384;
Gx = (float)(GyX-GyXo)/131;
Gy = (float)(GyY-GyYo)/131;
Gz = (float)(GyZ-GyZo)/131;
pitchgy = (Gy * ((float) dt/1000)) + pitch;
rollgy = (Gx * ((float) dt/1000)) + roll;
pitchac = atan2(Ax,Az) * (float)(180/PI);
rollac = atan2(Ay,Az) * (float) (180/PI);
roll = gain * rollgy + (1-gain)* rollac;
pitch = gain * pitchgy + (1-gain) * pitchac;
dt = millis()-t;
gyro_roll_input = roll;// feedback
pid_roll_setpoint = 0.0;
pid_i_mem_roll = 0.0;
pid_last_roll_d_error = 0.0;
pid();
sudut_servo= ((pid_output_roll+200)/400)*180;// dari besaran -
200~200 ke 0 ~ 180
A.write(sudut_servo);
B.write(sudut_servo);
Serial.print(" roll = ");
Serial.print(roll);
Serial.print(" error sudut = ");
Serial.print(pid_error_temp);
Serial.print(" servo = ");
Serial.print(sudut_servo);
Serial.print(" output = ");
Serial.println(pid_output_roll);
}

```

```

void pid(){
    //Roll calculations
    pid_error_temp = gyro_roll_input - pid_roll_setpoint; // EROR
    SUDUT TERHADAP 0.
    pid_i_mem_roll += pid_i_gain_roll * pid_error_temp; //
    PROPORSIONAL
    if(pid_i_mem_roll > pid_max_roll)pid_i_mem_roll = pid_max_roll; //
    I
    else if(pid_i_mem_roll < pid_max_roll * -1)pid_i_mem_roll =
    pid_max_roll * -1;
    pid_output_roll = pid_p_gain_roll * pid_error_temp + pid_i_mem_roll
    + pid_d_gain_roll * (pid_error_temp - pid_last_roll_d_error); // D
    if(pid_output_roll > pid_max_roll)pid_output_roll = pid_max_roll;
    else if(pid_output_roll < pid_max_roll * -1)pid_output_roll =
    pid_max_roll * -1;
    pid_last_roll_d_error = pid_error_temp;
}

```

Program potentiometer untuk mengendalikan motor brushless DC

```

#include <Servo.h>
Servo motor1;
Servo motor2;
void setup()
{
    Serial.begin(9600);
    motor1.attach(3);
    motor2.attach(5);
    motor1.writeMicroseconds(2300);
    delay(2000);
    motor1.writeMicroseconds(800);
    delay(6000);
    motor2.writeMicroseconds(2300);
    delay(2000);
    motor2.writeMicroseconds(800);
}

```

```

delay(6000);
}
void loop()
{
  int val1 = map(analogRead(0),0,1023,800,2300);
  int val2 = map(analogRead(1),0,1023,800,2300);
  motor1.writeMicroseconds(val1);
  motor2.writeMicroseconds(val2);
  Serial.print(val1);
  Serial.print(" ");
  Serial.print(val2);
  Serial.print("\n");
}

```

Program Sistem identifikasi Servo

a. Potentiometer

```

int analogPin = 0; // Means A0 input.If you use A1,
                  // then analogPin must be 1;
float angle;
// Initializes serial port.Baudrate is set to 9600.
int time,time2;
void setup()
{
  Serial.begin(9600);
}
// The main loop which Arduino runs.
void loop()
{
  time =millis();
  int ang = analogRead( analogPin ); // Reads A0 and gets W.
  angle = ang * (0.26392962);
  Serial.print(angle);
  Serial.print("      ");
}

```



```

time2 =millis()-time;
Serial.println(time2);
}

```

b. Motor servo

```
#include <VarSpeedServo.h>
```

```

VarSpeedServo myservo; // create servo object to control a servo
                        // a maximum of eight servo objects can be created

```

```

const int servoPin = 9; // the digital pin used for the servo
unsigned long time, time2;

```

```

void setup() {
  Serial.begin(9600);
  myservo.attach(servoPin); // attaches the servo on pin 9 to the servo
  object
  myservo.write(0,255,true); // set the intial position of the servo, as fast
  as possible, wait until done
}

```

```

void loop() {
  time =millis();
  myservo.write(10,100,true);    // move the servo to 180, max speed,
  wait until done
  delay(2000);
  time2 =millis()-time;
  Serial.println(time2);

  // write(degrees 0-180, speed 1-255, wait to
  complete true-false)
  myservo.write(0,100,true);    // move the servo to 180, slow speed,
  wait until done
  delay(3000);
}

```

Coding Matlab Root Locus Sistem

```

numg=[9.68];
deng=[1 0 -49.05];
numg1=[876.295];
deng1=[1 37.774 876.295];
G=tf(numg, deng);
G1=tf(numg1, deng1);
G2=G*G1
rlocus(G2)
poles=pole(G2)

```

Coding Matlab Root Locus PD

```

numg=[9.68];
deng=[1 0 -49.05];
numg1=[876.295];
deng1=[1 37.774 876.295];
numg2=[1 7.0036];
deng2=[1];
numg3=[1];
deng3=[1];
G=tf(numg, deng);
G1=tf(numg1, deng1);
G2=tf(numg2, deng2);
G3=tf(numg3, deng3);
G4=G*G1*G2*G3
rlocus(G4);
poles=pole(G4)

```

Coding Matlab Step Response PID dan PD

```

numg=[9.68];
deng=[1 0 -49.05];
numg1=[876.295];
deng1=[1 37.774 876.295];
numg2=[1 8.0036];      %D

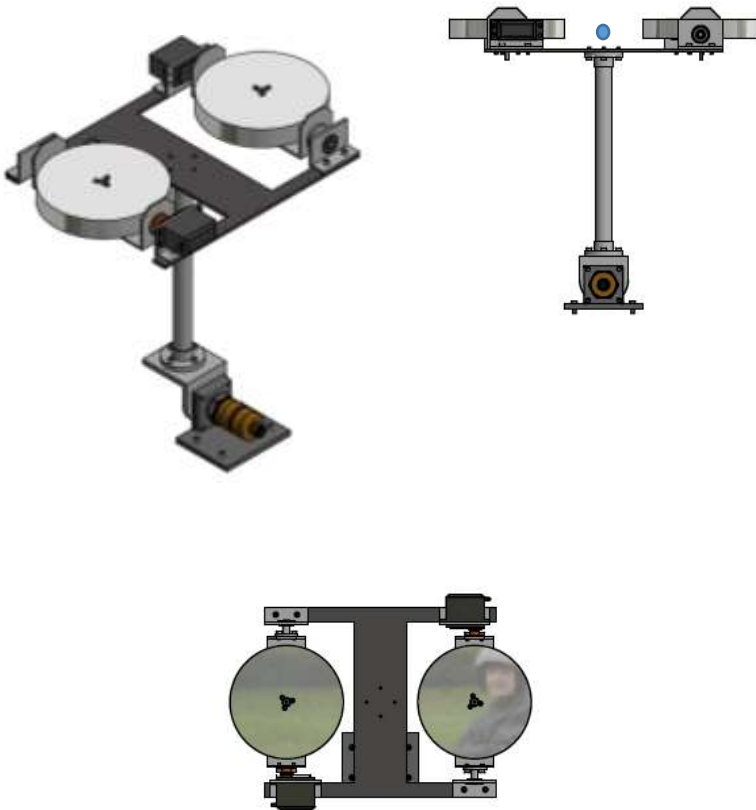
```

```

deng2=[1];
numg3=[1 0.8];           %I
deng3=[1 0];
numg4=[1.2];             %P
deng4=[1];
G=tf(numg, deng);
G1=tf(numg1, deng1);
G2=tf(numg2, deng2);
G3=tf(numg3, deng3);
G4=tf(numg4, deng4);
G5=G*G1*G2*G3*G4
T=feedback(G5,1);
poles=pole(T)
rlocus(T)
step(T);
numg5=[9.68];
deng5=[1 0 -49.05];
numg6=[876.295];
deng6=[1 37.774 876.295];
numg7=[1 8.0036];        %D
deng7=[1];
numg8=[1];               %I
deng8=[1];
G5=tf(numg5, deng5);
G6=tf(numg6, deng6);
G7=tf(numg7, deng7);
G8=tf(numg8, deng8);
G9=G5*G6*G7*G8
T1=feedback(G9,1,-1);
poles1=pole(T1)
rlocus(T1);
step(T,T1)

```

Gambar 3D dan gambar teknik *proi*



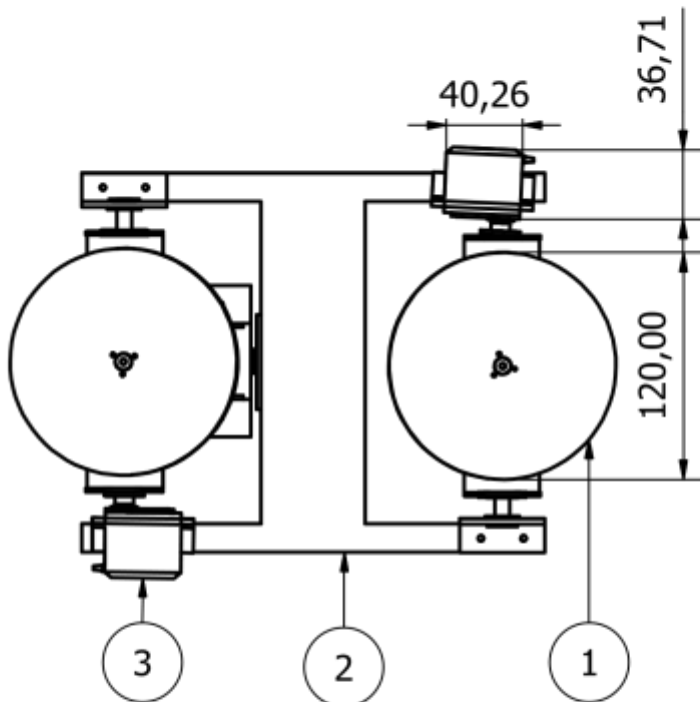
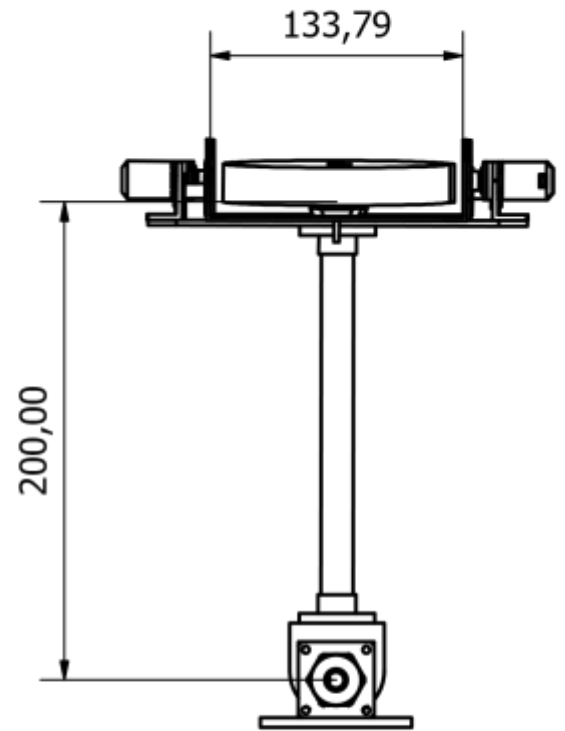
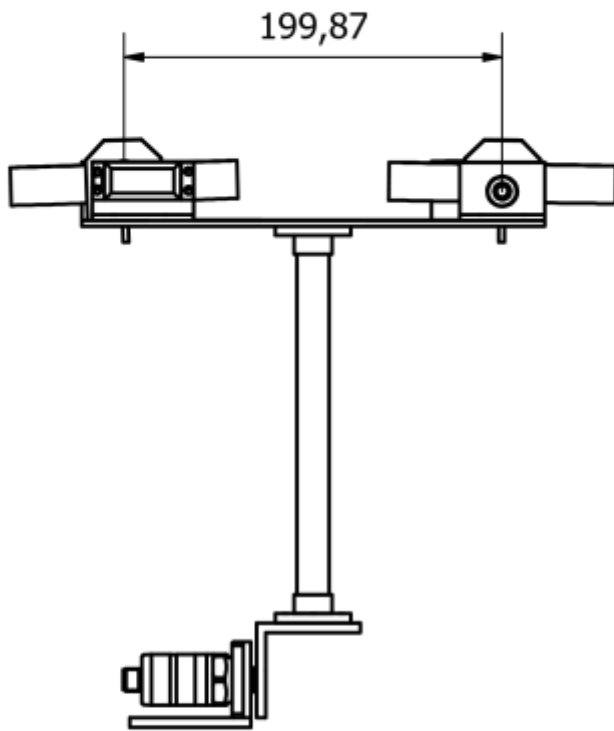
BIOGRAFI PENULIS



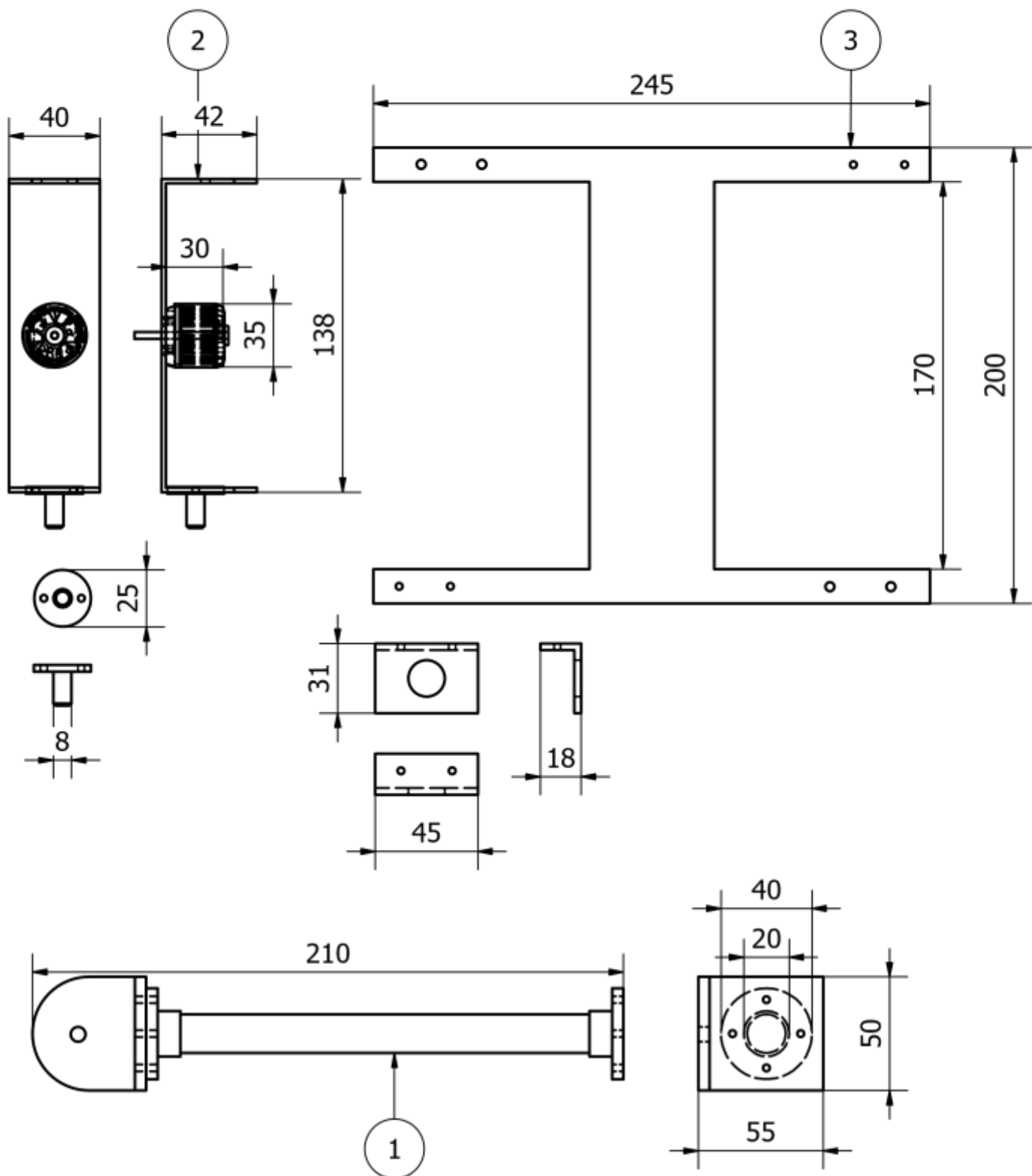
Moch Yusron Nafi', lahir di Kebumen, 10 Juli 1995. Penulis memulai jenjang pendidikan di sekolah dasar di RA Giwangretno pada tahun 2001 hingga 2004 dan SD Negeri 1 Karanggedang pada tahun 2004 hingga 2007. Penulis melanjutkan pendidikan tingkat menengah di SMP Negeri 3 Kebumen pada tahun 2007 hingga 2010. Kemudian penulis melanjutkan jenjang pendidikan di SMA Negeri 1 Kebumen pada tahun 2010 hingga 2013. Pada

tahun 2013, penulis memulai pendidikan strata 1 di jurusan Teknik Mesin, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Selama kuliah, penulis aktif mengikuti kegiatan robotika.

Email: muhammadyusronnafi@gmail.com



| | | | | | | |
|---|---------------|---------------------------|--|-------------|-------------|----|
| 3 | 2 | Servo | Metal gear | | | |
| 2 | 1 | Badan (Body) | Steel | | | |
| 1 | 2 | Flywheel Giroskop | Steel | | | |
| No. Bagian | Jumlah Bagian | Nama Bagian | Bahan | Normalisasi | Peringatan | |
| Kekasaran permukaan | | Toleransi ukuran dalam mm | | | | |
|  | | Skala : 1 : 5 | Digambar : Moch Yusron Nafi' | | Peringatan: | |
| | | Satuan panjang : mm | NRP : 2113100028 | | | |
| | | Tgl : 31/7/2017 | Dilihat 1 : Ari Kurniawan Saputra ST., MT. | | | |
| I T S | | | Inverted Pendulum | | 01 | A4 |



| | | | | | | |
|---|---------------------|---------------------------|--|-------------|-------------|----|
| 3 | 1 | H plate | Stainless steel | | | |
| 2 | 1 | Gimbal | Steel | | | |
| 1 | 1 | Rod | Steel | | | |
| No. Bagian | Jumlah Bagian | Nama Bagian | Bahan | Normalisasi | Peringatan | |
| Kekasaran permukaan | | Toleransi ukuran dalam mm | | | | |
|  | Skala : 1 : 2,5 | | Digambar : Moch Yusron Nafi' | | Peringatan: | |
| | Satuan panjang : mm | | NRP : 2113100028 | | | |
| | Tgl : 31/7/2017 | | Dilihat : Ari Kurniawan Saputra ST., MT. | | | |
| I T S | | | Inverted Pendulum | | 02 | A4 |